2011

Practical and Context-Aware Resource Adaptation in Mobile Networks

Xi Chen
Iowa State University

Practical and context-aware resource adaptation in mobile networks

by

Xi Chen

A dissertation submitted to the graduate faculty

in partial fulfillment of the requirements for the degree of

DOCTOR OF PHILOSOPHY

Major: Computer Engineering

Program of Study Committee:
Daji Qiao, Major Professor
Manimaran Govindarasu
Yong Guan
Aditya Ramamoorthy
Wensheng Zhang
John R. Schroeter

Iowa State University

Ames, Iowa

2011

DEDICATION

I would like to dedicate this dissertation to my wife Jing Lou, my parents and all my family members without whose support (both emotionally and financially) I would not have been able to complete this work. I would like to thank my advisor Daji Qiao for his great patience and strictness along my entire research path. I would also like to thank my colleagues and friends, who gave me a hand and shared the life with me during the last five years in the lovely town of Ames.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

ACKNOWLEDGEMENTS

I would like to take this opportunity to express my thanks to those who helped me with various aspects of conducting research and the writing of this dissertation. First and foremost, Dr. Daji Qiao for his guidance, patience and support throughout this research and the writing of this dissertation. His insights and words of encouragement have often inspired me and renewed my hopes for completing my graduate education. I would also like to thank my collaborators without whom this work would not have been possible: Sunggeun Jin, Jeonggyun Yu, Prateek Gangwal, Yue Zhao and Brian Peck. I would additionally like to thank my friends in Ames for the happiness they bring to me: Jing Wang, Chang Liu, Lei Ke, Wei Lu, Wei Zhou, Guanqun Yang and Ka Yang.

ABSTRACT

With the proliferation of various portable devices such as smart phones, netbooks and tablets, it becomes more important to design and implement effective resource management schemes with (i) the increasing number of users in the network and (ii) the expectation of frequent and fast mobility of network users. In this dissertation, we conclude that the key to solve the problem in mobile networks is adaptive resource allocation, which requires the system to behave in an adaptive manner considering the dynamic network conditions and various context of mobile users. Specifically, we study the following critical resource allocation issues in this dissertation: (i) rate adaptation; (ii) station handoff; (iii) load balancing; and (iv) power saving, for each we have proposed an adaptive scheme, implemented it in the MadWifi device driver, and demonstrated its effectiveness via experiments.

CHAPTER 1. INTRODUCTION

1.1 Adaptive Resource Allocation in Mobile Networks

The wireless industry is in the midst of a fundamental shift from providing voice and best-effort data services to offering customers an array of real-time multimedia services, including a wide variety of audio, video and data communications capabilities. Future wireless networks will be integrated into every aspect of daily life, and therefore could affect our life in a magnitude similar to that of the Internet and cellular phones. Portable devices, such as smart phones, netbooks and personal multimedia devices, even vehicles equipped with communication devices are becoming increasingly popular in recent years. Fig. 1.1 shows an example of such networks and its possible applications. In this figure, we illustrate the access point (AP) deployment nearby Coover Hall and Atanasoft Hall in Iowa State University campus. In this example, static or mobile users obtain data services via indoor or outdoor APs. Mobile users with portable devices also can exchange their data in an ad-hoc manner such as u_1 and u_2.

As many new applications for wireless devices emerge which require better mobility support, it has posted great challenges to researchers in the wireless networking area. Some of the challenges involve the mapping of geographic coordinates to network addresses, the intermittent connectivity of mobile users, the transparency to upper-layer applications and the route construction to a moving destination. One of the fundamental challenges is how to allocate resources in wireless networks to diverse mobile users who may experience different channel conditions at different times and locations, and may have different demands and capabilities. The key to address this challenge is the need of *adaptive* resource allocation among mobile users, since resource allocation in mobile networks typically requires the system to behave in an adaptive manner considering the dynamic network conditions, and the stations to be context-aware (e.g., be aware of channel condition, traffic load, distance, velocity, moving trajectory, etc.) so as to make appropriate allocation decisions.

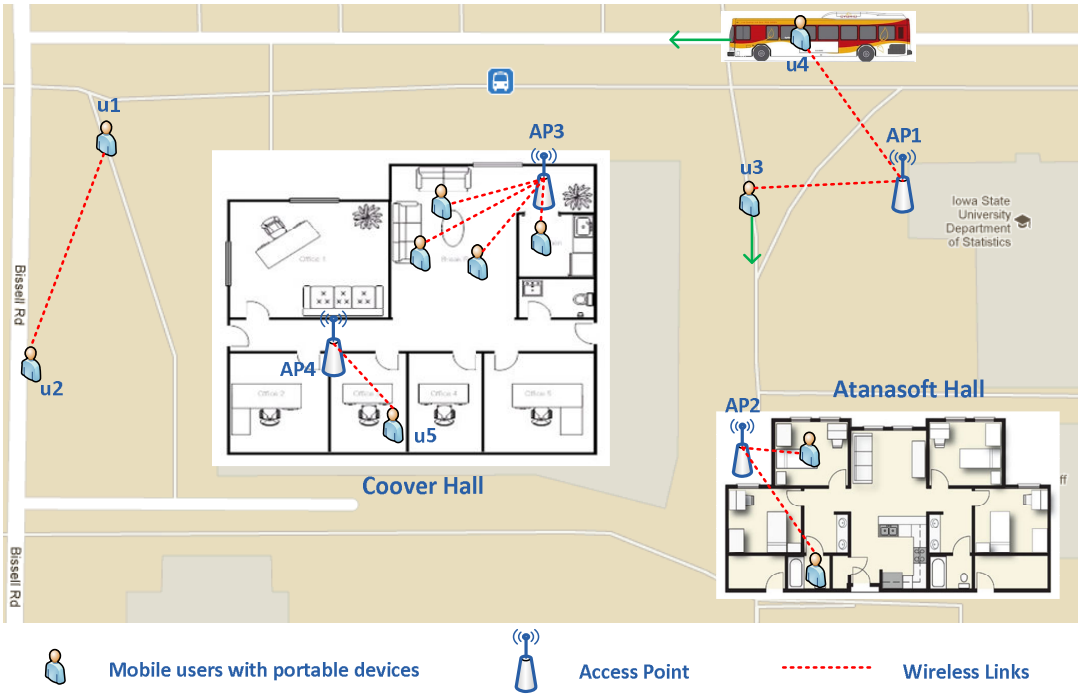


Figure 1.1 An example mobile network and its applications

The goal of adaptive resource allocation in mobile networks is to guarantee a satisfactory system performance in terms of (i) aggregate system throughput; (ii) power consumption of mobile devices; (iii) end-to-end packet delay; (iv) throughput fairness among mobile users; and (v) service outages. To design a sophisticated and adaptive system to achieve the above goal, the following procedures are usually necessary:

- Collect dynamic context information for each station, e.g., channel condition, transmission rate, frame size, traffic load, velocity, etc.. These context information could be obtained by either background analysis from wireless traffic or active probing mechanism. In certain cases, external sensors such as GPS sensor could be adopted to report such information as well.

- Propose adaptive algorithm to optimize the system performance, with the collected context information as system inputs. In this step, stations might need to collaborate with each other to make more appropriate decisions. For a practical solution, the computation overhead of the algorithm should be optimized since the algorithm is supposed to be performed periodically over

the runtime.

- Adjust the behavior of the system according to the output of the adaptive algorithm. These actions include but not limited to (i) updating the transmission rate; (ii) adjusting the association pattern; (iii) controlling the transmission power; and (iv) making sleep/wakeup schedules.

1.2 Problem Identification

As discussed above, adaptive resource allocation in mobile networks is a non-trivial research topic. A variety of fundamental problems need to be solved before a large-scale deployment of the system may be carried out. In this dissertation, the following topics will be discussed:

- *Rate adaptation* is one of the fundamental resource allocation issues, and the transmitter should adapt its physical layer transmission rate according to time-varying link conditions so as to maximize the link level throughput. The challenge lies in that (i) the transmitter should accurately estimate the channel condition and respond quickly especially in high-speed mobile environments; and (ii) the transmitter should identify the existence of hidden nodes in the network and take corresponding actions. All wireless links in a mobile network could benefit from a well-designed rate adaptation scheme.

- *Station handoff* is another interesting resource allocation problem. For example in Fig. 1.1, the outdoor mobile user u_3 should handoff from AP_1 to AP_2 if its current link condition with AP_1 becomes unacceptable. Similarly, the user u_4 on the CyRide bus needs to handoff to AP_3 that may provide better service when he moves away from AP_1. How to reduce the handoff delay and how to make appropriate handoff decisions are two fundamental challenges in designing an effective handoff scheme. An ideal handoff solution needs to be adaptive to the dynamic link conditions, different AP loads as well as other dynamics and context information available in the network.

- *Load balancing* is becoming more and more important nowadays with the emergence of various portable devices and user applications. For example in Fig. 1.1, all APs serve multiple users, each with various applications running on their portable devices. Since user traffic patterns are

quite dynamic, it is highly possible that the load of APs are completely unbalanced, which leads to service degradation on the more saturated AP. As illustrated in Fig. 1.1, AP_3 serves many more users and is more saturated than AP_4, in which case some users can be switched to AP_4 in order to balance the loads and achieve better system throughput. With smart load balancing scheme, APs should collaborate and switch the user dynamically to achieve better system-wide performance. The fundamental challenges of load balancing are (i) how to determine stations that should be switched and (ii) how to switch the station with less overhead if station reassociation is frequent.

- *Power saving* tries to minimize the power consumption of mobile devices while maintaining a satisfactory throughput performance. For example in Fig. 1.1, user u_5 sleeps for most of the time if there is only few traffic destined to him. By putting the device into sleeping adaptively, the power saving scheme strikes a balance between the end-to-end packet delay and the battery life. With a target packet delay level, a good power saving scheme should sleep as much as possible to prolong the battery life. An ideal solution should consider (i) the current context information, e.g., link condition, traffic load; as well as (ii) the predicted future context information, in order to make better sleep/wakeup decisions.

1.3 Proposed Research

In this dissertation, we study resource adaptation problems in IEEE 802.11-based [1] mobile networks. We aim at identifying the critical issues that affect the performance of such networks and propose *practical* solutions that can be implemented with off-the-shelf IEEE 802.11 devices and offer greater performance in terms of system throughput, end-to-end packet delay, etc.. The following research topics are included in this dissertation:

- *Prompt and Effective Rate Adaptation.*

  We study the problem of how to adaptively adjust the transmission rate to maximize the link level throughput especially in mobile environments such as vehicular networks. Besides, we also propose practical solution to solve the hidden node problem and keep it compatible with

legacy stations, which is critical for a real world deployment. We perform real world experiments and observe some unique issues in high-speed mobile environments. To deal with these issues, we propose a practical rate adaptation scheme called RAM (Rate Adaptation in Mobile environments) and implement it in the MadWifi device driver.

- *Fast and Seamless Handoff.*

  We study the problem of how to perform fast and seamless handoff in IEEE 802.11 networks by taking into account system throughput, packet delay as well as service outage. By investigating the components of handoff delay, we discuss how to reduce the handoff delay in order to minimize the interrupt to normal user communication. Meanwhile, we also study how to make appropriate handoff decisions with the goal of increasing the system performance. Extensive experiments have been performed and some observations have been made based on which we propose a unique fast handoff scheme called HaND (Handoff with Null Dwell time), which tries to eliminate the channel dwell time hence reducing the handoff delay, and enforces the system to make more appropriate handoff decisions based on the collected context information.

- *Seamless Load Balancing with Multiple Interfaces.*

  The AP with multiple wireless interfaces are more and more popular in the current market, however the advantage of such a One-AP-Multiple-Interface (OAMI) architecture is usually not fully utilized. In this work, we consider the popular OAMI AP and propose a seamless load balancing scheme called SAP (Smart Access Point), in order to providing adequate Wi-Fi services to meet user demand in densely populated environments. Specifically, SAP balances the load via (i) adjusting the association scheduling decisions between multiple interfaces dynamically based on stations' actual and varying traffic load conditions; and (ii) executing the association scheduling decisions with seamless handoff of stations between interfaces with zero handoff delay. Similar to our previous work, we implemented SAP in the MadWifi driver and demonstrated its effectiveness via experiments.

- *Mobility-Aware Power Save Mode*

  Power Save Mode (PSM) is the default power management scheme in the 802.11 standard, which

allows the station to sleep for most of the time and only wake up periodically to receive packets. PSM is originally designed to reduce the power consumption of a wireless station when it is lightly-loaded. In this work, we explore context information such as the transmission rate and mobility pattern to further enhance the performance of PSM. As we will show, non-negligible performance gain could be achieved by adaptively tuning the behavior of PSM with the help of the collected context information of mobile stations.

The rest of the dissertation is organized as follows. In Chapter 2, we present our research on rate adaptation in mobile environments. A new fast handoff architecture for IEEE 802.11 networks is proposed and presented in Chapter 3. The smart AP with seamless load balancing multiple interfaces is discussed in Chapter 4. Chapter 5 presents our mobility-aware Power Save Mode scheme. Chapter 6 concludes this dissertation with a summary of the main contributions and discusses possible future research topics.

CHAPTER 2. PROMPT AND EFFECTIVE RATE ADAPTATION

2.1 Introduction

The increasing number of IEEE 802.11 devices have been used in various mobile applications. Since most resource management schemes for 802.11 devices are designed for static environments, 802.11-based systems may experience severe performance degradation in mobile environments, such as low throughput and high latency. This is due to the salient differences in wireless channel characteristics between static and mobile environments. For example, channel conditions in mobile environments usually exhibit more severe asymmetry and higher fluctuation than those in static environments.

Rate adaptation is one of the fundamental resource allocation issues for 802.11 devices. The goal is to maximize the throughput via exploiting the multiple transmission rates available for an 802.11 device and adjusting its transmission rate dynamically to the time-varying and location-dependent wireless channel condition. To deal with the channel asymmetry problem, the channel estimation is better to be done at the receiver side which monitors the channel quality and makes the rate selection for the next frame transmission. However the challenge to such receiver-based scheme is how to feed the selection back to the transmitter. Most receiver-based schemes fulfill this by modifying the MAC layer frame formats and hence are incompatible with the 802.11 standard. Moreover, these receiver-based schemes can not be implemented with off-the-shelf devices since they need to modify the device firmware completely. To deal with the high channel fluctuation, a good rate adaptation scheme should be able to predict the channel condition correctly without any over-estimation or under-estimation. In addition, a good rate adaptation scheme should identify the existence of hidden nodes in the network and take corresponding actions.

2.2 Literature Survey

Rate adaptation in static environments has been well-studied in the past [2–15]. As shown in Table 2.1, these rate adaptation schemes can be classified in the following ways: *transmitter-based* or *receiver-based*; *packet statistics-based* or *SNR-based*; *window-based* or *frame-based*.

Transmitter-based vs. receiver-based. In *transmitter-based* schemes, the transmitter makes the rate selection decisions. By comparison, in *receiver-based* schemes, the receiver monitors the channel quality, makes the rate selection for the next frame transmission and feeds the decision back to the transmitter.

Packet statistics-based vs. SNR-based. Based on the information used to infer the channel condition, rate adaptation schemes can be classified as *packet statistics-based* and *SNR-based*. In packet statistics-based schemes, ARF [2], AARF [3] (also known as AMRR for its implementation in the MadWifi device driver) and CARA-like schemes [4] [5] use consecutive frame transmission failure and success counts as indicators of the channel quality. ONOE [11] and RRAA [6] calculate the frame loss ratio and RRAA compares it with certain thresholds to make rate updating decisions. SampleRate [7] chooses the rate with the shortest expected frame transmission time. In SNR-based schemes, CHARM [8], the scheme described in [9] and SGRA [10] are transmitter-based. They use the RSSI (Receive Signal Strength Indictor) values of the ACK frames received by the transmitter to infer the channel condition at the receiver side based on the assumption of a symmetric channel. In comparison, RBAR [12], OAR [13] and RARA [14] are receiver-based and they instead use the RSSI values of data frames

Table 2.1 Classification of Existing Rate Adaptation Schemes

Schemes	Tx/Rx-based	Based on SNR	Window/Frame-based	Implemented	Deal with hidden node
ARF/AARF	Tx	No	window	Yes	No
CARA-like	Tx	No	window	No	Yes
RRAA	Tx	No	window	Yes	Yes
SampleRate	Tx	No	window	Yes	No
CHARM	Tx	Yes	frame	Yes	No
Scheme in [9]	Tx	Yes	frame	No	No
SGRA	Tx	Yes	frame	Yes	No
ONOE	Tx	No	window	Yes	No
RBAR	Rx	Yes	frame	No	No
OAR	Rx	Yes	frame	No	No
RARA	Rx	Yes	frame	No	No
RAF	Rx	Yes	frame	No	Yes

received by the receiver. In RAF [15], the receiver determines the optimal transmission rate and frame size based on the measured interference level.

Window-based vs. frame-based. Based on the rate updating period, rate adaptation schemes can be classified as *window-based* and *frame-based*. ARF, AARF and CARA-like schemes use frame transmission failure and success counts and make rate adjustment when the number of frame transmission failures or successes is above a certain threshold. The window sizes for RRAA and SampleRate are 150 ms and one second respectively by default. Window-based schemes are reactive in nature as they rely on the past history to predict future channel conditions. Moreover, it is usually difficult to determine the optimal window size in dynamic environments when the channel condition varies often. In comparison, frame-based schemes adapt much faster to rapid variations of the channel condition that are often caused by fading and mobility.

In Table 2.1, we also list whether a rate adaptation scheme has already been implemented. To our knowledge, none of the receiver-based schemes has yet been implemented to work with commercial 802.11 devices. In fact, these receiver-based schemes likely are not implementable with commercial 802.11 devices for the following reasons. RBAR and OAR require modifications to the CTS (and possibly RTS) frame formats, which does not conform to the 802.11 standard, while the variation patterns of the ACK transmission rate proposed in RARA are not supported by commercial 802.11 devices and their device drivers.

In the presence of hidden nodes, it is difficult for the transmitter to differentiate channel-error-induced frame transmission failures from collision-induced ones, which may lead to pessimistic usage of the transmission rates. Adaptive usage of RTS/CTS has been recognized as an effective way to deal with hidden nodes, and it has been used in a few rate adaptation schemes such as CARA-like schemes and RRAA.

The scheme in [16] combines the sender- and receiver-based methods to adjust the rates of data as well as control frames. In [17], the authors propose a rate adaptation scheme for vehicular networks based on the context information such as distance and relative velocity. It is a history-based approach and requires repetitive training before usage. It is designed specifically for vehicles traveling along known routes. Therefore, this scheme may not be suitable for dynamic mobile environments where

routes are not known *a priori* and the channel condition is unpredictable. In [18], the authors modify SampleRate for mobile environments by reducing the estimation window size.

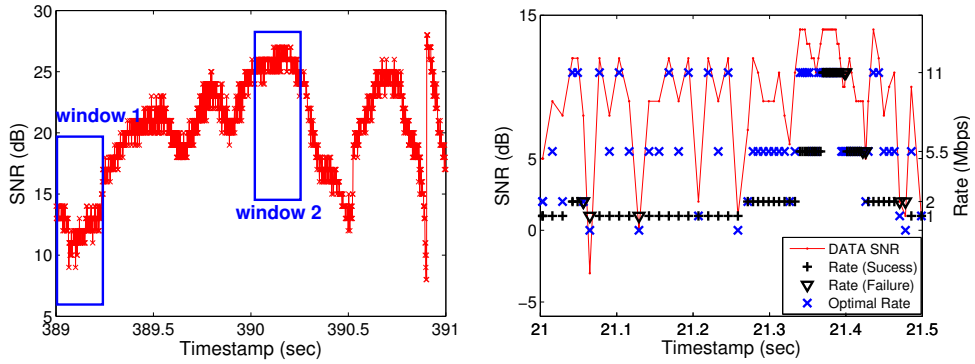
2.3 Observations & Motivations

To design an effective rate adaptation scheme for mobile environments, it is critical to have a good understanding of the characteristics of wireless channels in mobile environments. To do so, we conduct experiments with two laptops equipped with CB9-GP-EXT 802.11a/b/g cards [19] in various indoor (static and mobile) and outdoor (vehicular) environments. Each laptop is loaded with the MadWifi device driver v0.9.4 [20] to measure and record the channel conditions. Details of the setup of the experiments can be found in Section 2.5.1. In this section, we present the observations and findings from the experiments.

2.3.1 Issues with Packet Statistics-based Schemes

2.3.1.1 Window-based rate adaptation schemes

This type of schemes collect packet statistics within a time window (or a window of certain number of packets) and make rate selection decisions at the end of the window. In mobile environments, since the channel condition fluctuates frequently, which will be discussed in Section 2.3.3, packet statistics collected at the current window may become obsolete when making rate selection decisions for future transmission attempts. Fig. 2.1(a) shows a trace of DATA SNR values in an experimental run for an outdoor vehicular scenario. It can be seen from the figure that it would be too pessimistic or optimistic to use the packet statistics collected from window 1 or window 2 to select rates for future packet transmissions. Another issue with window-based rate adaptation schemes lies in the selection of a proper window size. If the window size is too large, some of the collected information may become outdated at the end of the window. In comparison if it is too small, the collected statistics may not be accurate enough.



(a) (b)

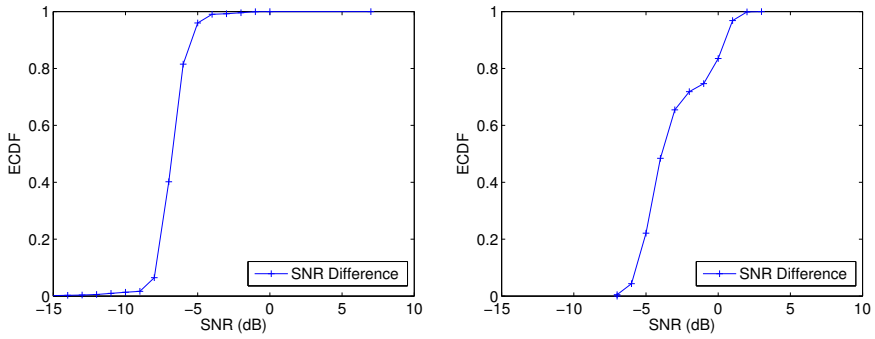
Figure 2.1 Packet statistics-based rate adaption schemes are not suitable for mobile environments. (a) Issues with window-based schemes. (b) Issues with count-based schemes.

2.3.1.2 Count-based rate adaptation schemes

ARF and CARA-like schemes use consecutive frame transmission failure or success counts to select the rate for the next transmission attempt. They increase the rate after 10 consecutive transmission successes and decrease the rate upon two consecutive failures. This type of approaches may not work effectively in mobile environments with high SNR fluctuations. As shown in Fig. 2.1(b), the success count of 10 may be too conservative for rate increasing, and the failure count of two may be too pessimistic for rate decreasing, which may lead to potential under-utilization of the channel.

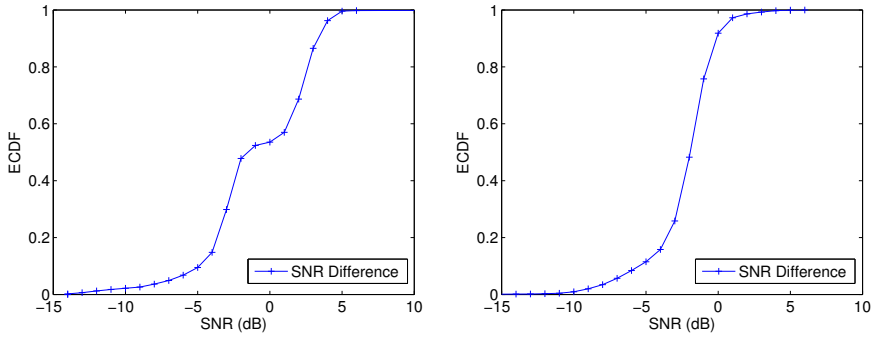
2.3.2 Severe Channel Asymmetry

One interesting observation from our experiments is the severe channel asymmetry in practical scenarios. As shown in Figs. 2.2 and 2.3, ACK SNR values collected at the transmitter usually differ significantly from DATA SNR values collected at the receiver. The SNR difference is as high as 15 dB in some outdoor vehicular scenarios. Since channel symmetry is one of the key assumptions in several existing transmitter-based rate adaptation schemes such as CHARM and SGRA, these schemes may not be suitable for mobile environments. Instead, receiver-based approaches may be a better option.



(a) Indoor static scenario. (b) Indoor mobile scenario.

Figure 2.2 ECDF (Empirical Cumulative Distribution Function) of the difference between DATA and ACK SNR values in indoor static and mobile scenarios.



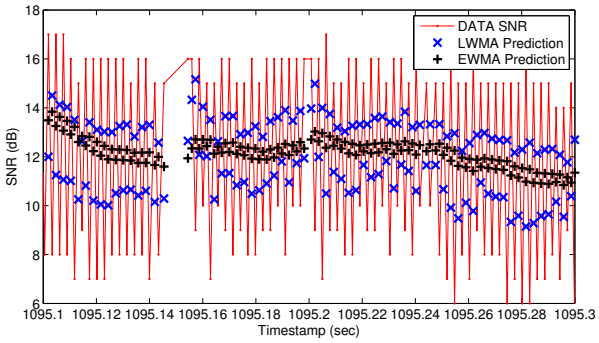
(a) Outdoor vehicular scenario 1. (b) Outdoor vehicular scenario 2.

Figure 2.3 ECDF of the difference between DATA and ACK SNR values in vehicular scenarios.

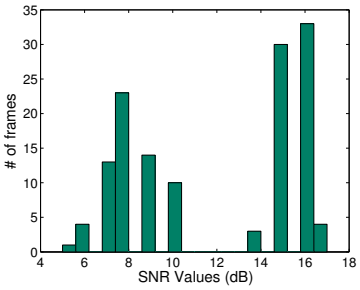
2.3.3 High SNR Fluctuation

High SNR fluctuation is another important observation from our experiments. In some traces such as the one shown in Fig. 2.4(a), the differences between consecutive SNR values are as large as 10 dB. From the experiments, we notice that high SNR fluctuation usually occurs when the environment suddenly changes, e.g., opening or closing a door, sudden acceleration of the vehicle, and vehicle making a turn. Fig. 2.4(b) plots the histogram of the SNR values shown in Fig. 2.4(a) and we can see that the distribution of SNR values is quite irregular.

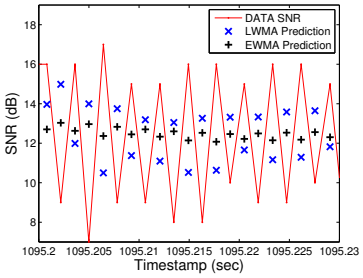
SNR prediction algorithms used in existing rate adaptation schemes may not be able to handle the high SNR fluctuation properly. As shown in Figs. 2.4(a) and 2.4(c), the Light Weighted Moving Average (LWMA) scheme used in CHARM almost always uses the previous SNR value as the prediction



(a) SNR values and predictions by LWMA and EWMA.



(b) Histogram of SNR values.



(c) Zoomed view of (a).

Figure 2.4 High SNR fluctuation in mobile environments.

for the next SNR value, which results in a large number of over-estimations of SNR values and hence frame transmission failures. Similar problem exists for the simple Exponentially Weighted Moving Average (EWMA) scheme as well.

2.4 Design and Implementation of RAM

To deal with the issues discussed in the previous section, we propose a practical rate adaptation scheme, called RAM (Rate Adaptation in Mobile environments) that can be implemented with commercial 802.11 devices. As shown in Fig. 2.5, RAM has the following components – at the receiver side: (i) SNR prediction (ii) rate selection based on SNR prediction (iii) feedback of rate selection to the transmitter; and at the transmitter side: (i) rate updating and (ii) adaptive usage of RTS/CTS. Interoperability between RAM-based and legacy 802.11 devices will be discussed in the end.

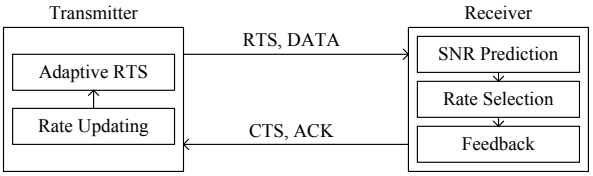


Figure 2.5 Overall structure of RAM.

2.4.1 Receiver: SNR Prediction

To deal with high SNR fluctuation and irregular SNR distribution, we propose a simple conservative SNR prediction algorithm as follows. It maintains the moving averages of the SNR values as well as the deviations to the average SNR value:

$$\begin{cases} S_{\text{avg}} = (1-\delta)\cdot S_{\text{avg}} + \delta \cdot S_{\text{curr}}, \\ \\ DEV_{\text{avg}} = (1-\rho)\cdot DEV_{\text{avg}} + \rho \cdot |S_{\text{curr}} - S_{\text{avg}}|, \end{cases} \tag{2.1}$$

and predicts the SNR value for the next frame as:

$$S_{\text{est}} = S_{\text{avg}} - \eta \cdot DEV_{\text{avg}}, \tag{2.2}$$

where S_{curr} is the SNR value reported by MadWifi upon each frame reception and δ, ρ, η are design parameters. Ideally, the SNR value of each received frame should be used as S_{curr} in the algorithm. Unfortunately, the current MadWifi does not support per-frame-based SNR measurement. While it measures the received signal level for each frame, it only updates the noise level upon each interrupt and usually multiple frames are served between interrupts [21]. Therefore, strictly speaking, the SNR value reported by MadWifi upon each frame reception is *not* the exact SNR value of the frame *but* an approximation to it. Nevertheless, even with such limitation of the current MadWifi, RAM still yields a noticeable performance improvement over existing rate adaptation schemes, which will be shown in later sections via both experimental and simulation results.

In comparison, both EWMA and LWMA predict the SNR value without considering the deviation of recent SNR values. They work as follows:

$$S_{\text{est}} = S_{\text{avg}} = (1-\delta)\cdot S_{\text{avg}} + \delta \cdot S_{\text{curr}}. \tag{2.3}$$

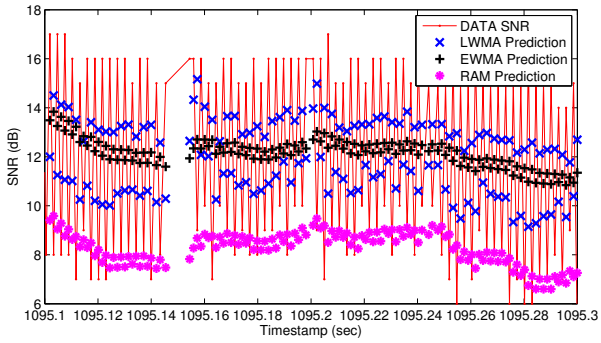


Figure 2.6 An example of our SNR prediction algorithm.

For EWMA, δ is a fixed smoothing factor. For LWMA, δ is adjusted during the runtime: $\delta = \frac{1}{1+f(T)}$ where T is the time interval between consecutive transmissions and $f(T)$ is a linearly decreasing function of T, starting at one and decreasing to zero when T exceeds a decision time window (two seconds).

By considering the deviation of recent SNR values when making the prediction, our algorithm can deal with high SNR fluctuation well. It is designed to predict future SNR values as accurately as possible without over-estimating them. This can be seen from an example shown in Fig. 2.6 where the predictions by our algorithm follow the lower envelop of the SNR variation closely.

2.4.2 Receiver: Rate Selection based on SNR Prediction

To select the proper rate for the next frame transmission to maximize the throughput, RAM maintains a throughput-vs-(rate, SNR) table. For each (rate = R, SNR = S) pair in the table, we use $G(R, S)$ to denote the expected throughput when the frame is transmitted at rate R and its SNR at the receiver is S. The table is updated upon each data frame reception. In order to respond quickly in the presence of bursty strong interference in the network, our table updating scheme uses a moving average approach:

$$G_{\text{avg}}(R, S) = (1 - \gamma) \cdot G_{\text{avg}}(R, S) + \gamma \cdot G_{\text{inst}}(R, S), \tag{2.4}$$

where $G_{\text{inst}}(R, S)$ is the instant throughput for each distinct (R, S) pair observed between two consecutive data frame receptions. Then, based on the predicted SNR value (S_{est}), the receiver looks up the table and selects the rate for the next frame transmission as follows:

$$R^* = \arg\max_R G_{\text{avg}}(R, S_{\text{est}}). \tag{2.5}$$

We implement RAM in the MadWifi device driver, which employs the *multi-rate retry mechanism* to transmit a data frame. In RAM, transmitter and receiver agree on the multi-rate retry mechanism and receiver may use this knowledge to update the throughput-vs-(rate, SNR) table. Before proceeding to the details about how the table is updated, we first give a brief introduction about the multi-rate retry mechanism.

2.4.2.1 Multi-rate Retry Mechanism

In MadWifi, whenever a frame is ready to send, MadWifi can specify up to four different rates ($r_1 > r_2 > r_3 > r_4$) along with their maximum retry counts ($c_i,\ i = 1, \cdots, 4$) for the frame and pass these information to the card firmware along with the frame. The frame is discarded after $(c_1 + c_2 + c_3 + c_4)$ unsuccessful transmission attempts, i.e., c_i attempts at the rate of r_i. In MadWifi, the maximum total number of transmission attempts for a frame is 10, meaning that $(c_1 + c_2 + c_3 + c_4) \leqslant 10$. The card firmware reports the total number of transmission attempts to MadWifi after the frame has been transmitted successfully or discarded. In RAM, we set $c_1 = 4$, $c_2 = c_3 = c_4 = 2$, and r_{i+1} to be the next lower rate to r_i ($i = 1, 2, 3$). The reason for setting a larger retry count ($c_1 = 4$) for the first rate r_1 is because RAM adopts a conservative SNR prediction algorithm and hence r_1 usually has already been selected conservatively.

2.4.2.2 Updating the Throughput-vs-(Rate, SNR) Table

When the receiver receives a data frame successfully, there are two possible outcomes for each of the unsuccessful transmission attempts (if any) prior to the successful reception of the frame: *frame was corrupted but header can be retrieved by the receiver*, or *frame was completely lost*. For the former case, the information about the unsuccessful transmission attempt such as the transmission rate and the SNR of the frame are reported to MadWifi by the card firmware, while for the latter case, these information are not available. Interestingly, our experiments show that the latter case rarely occurs in practice and we hence always assume the former case when updating the throughput-vs-(rate, SNR) table. Note that the SNR values reported by the card firmware are integer values, meaning that the number of entries in the table is a small finite number.

Suppose a data frame is received successfully during the k-th attempt at rate R_k and SNR of S_k, where $1 \leqslant k \leqslant (c_1 + c_2 + c_3 + c_4)$. For each of the unsuccessful attempts, the receiver retrieves the rate and SNR information reported by the card firmware. Then, we have the following:

$$\begin{cases} L(R_j, S_j) = 0, & \text{for } 1 \leqslant j < k, \\ \\ L(R_j, S_j) = \textit{data\_payload}, & \text{for } j = k, \end{cases} \tag{2.6}$$

and

$$T(R_j, S_j) = \textit{txtime}(R_j) + \textit{backoff}(j), \quad \text{for } 1 \leqslant j \leqslant k, \tag{2.7}$$

where $L(R_j, S_j)$ and $T(R_j, S_j)$ are the amount of data received successfully at rate R_j and SNR of S_j, and the transmission time for such a frame, respectively. Here, *data_payload* is the data payload of the frame, *txtime*(R_j) is the frame transmission duration at rate R_j, and *backoff*(j) is the average backoff time prior to the j-th transmission attempt, which is given by:

$$\textit{backoff}(j) = \min\left[\frac{(\mathrm{CW}_{\min} + 1) \cdot 2^{j-1} - 1}{2}, \frac{\mathrm{CW}_{\max}}{2}\right] \times \text{aSlotTime}. \tag{2.8}$$

Then, the instant throughput for each distinct (R, S) pair observed during k transmission attempts is calculated by

$$G_{\text{inst}}(R, S) = \frac{\sum_{(R,S)==(R_j,S_j)} L(R_j, S_j)}{\sum_{(R,S)==(R_j,S_j)} T(R_j, S_j)}, \tag{2.9}$$

and the corresponding $G_{\text{avg}}(R, S)$ is updated using (2.4).

2.4.3 Receiver: Feedback of Rate Selection to the Transmitter

The 802.11 standard [1] specifies that the ACK frame should be transmitted at the highest rate in the basic rate set that is less than or equal to the transmission rate of the data frame it is acknowledging. We call such ACK transmission rate the *default ACK rate*. For example, the 802.11g basic rate set is {1, 2, 5.5, 11, 6, 12, 24} Mbps. So if a data frame is transmitted at 18 Mbps, the default rate of the corresponding ACK frame is 12 Mbps. In practice, MadWifi allows two different transmission rates for ACK frames, as listed in Table 2.2 for Atheros chipset-based 802.11g cards. MadWifi can specify that an ACK frame is transmitted at a low rate or a high rate (the default rate) via setting different values of a special register [22].

Table 2.2 In MadWifi: Two Rates Available for ACK Frames

data rate (Mbps)	1	2	5.5	11	6	9	12	18	24	36	48	54
low ACK rate	1	2	2	2	6	6	6	6	6	6	6	6
high ACK rate	1	2	5.5	11	6	6	12	12	24	24	24	24

RAM takes advantage of this MadWifi feature and conveys the feedback information implicitly via the ACK transmission rate variation. Specifically, if the receiver wants to inform the transmitter to send the next frame at the same rate as the previous frame, or at the next higher rate, it transmits the ACK frame at the default high rate or the low rate, respectively. For example, if the receiver receives a data frame successfully at 36 Mbps, it can signal the transmitter to send the next frame at 36 or 48 Mbps by transmitting the ACK frame at 24 or 6 Mbps, respectively. RAM-like rate adaptation solutions may be designed for other wireless communication devices as long as the device driver supports similar functions as MadWifi, such as dual ACK transmission rates.

Note that for rates of 1, 2, 6, and 9 Mbps, there is only one option for the ACK transmission rate. In RAM, we disable the data transmission rates of 6 and 9 Mbps since it has been observed from experiments that the throughput performances of 6 and 9 Mbps are worse than that of 5.5 Mbps [8]. For rates of 1 or 2 Mbps, rate increasing decisions are made at the transmitter side, which will be explained in the next section, along with rate decreasing decisions which are always made at the transmitter side regardless of the current rate.

2.4.4 Transmitter: Updating the Transmission Rate

As described in Section 2.4.2.1, once r_1 is decided for a data frame, the multi-rate retry mechanism for the frame is decided. In RAM, we decide r_1 for the next data frame according to the transmission result of the *last* attempt (suppose at the rate of R_{last}) of the previous data frame:

1) If it fails: the transmitter sets r_1 to R_{last}.

2) If it succeeds: the transmitter may take the following actions depending on R_{last}:

- If $R_{\text{last}} > 2$ Mbps, the transmitter relies on the feedback from the receiver to set the rate for the next frame. Specifically, if the transmitter receives an ACK frame at the default high rate, $r_1 = R_{\text{last}}$; otherwise, r_1 is set to the next higher rate to R_{last}.

- If $R_{\text{last}} = 1$ or 2 Mbps, the transmitter makes the rate updating decision using the following heuristic. In RAM, the transmitter keeps track of the ACK SNR values. When the current ACK SNR is 5 dB higher than the previous one or when the number of consecutive frame transmission successes is larger than four, r_1 is set to the next higher rate to R_{last}. In an extreme case when $R_{\text{last}} = 1$ Mbps and the current ACK SNR is 9 dB larger than the previous one, r_1 is increased to 5.5 Mbps directly. These thresholds are obtained from the experiments.

By default, MadWifi uses the high ACK rate to calculate the NAV value for a data frame transmission. In RAM, since the receiver may transmit an ACK frame at the low rate to signal rate increasing for the next data frame, we modify the NAV calculation in MadWifi by using the low ACK rate intead. This can be done by modifying the value of a special register [22]. Since ACK frames are short, such modification does not affect the performance much, which will be discussed in Section 2.6.3. In addition, since the RAM receiver uses a moving average to update the SNR estimation and feeds back the rate selection decisions to the transmitter on a per-frame basis, the RAM transmitter is able to converge quickly (usually a few frames) to the proper transmission rate even when the interval between two consecutive frame transmissions is large.

2.4.5 Transmitter: Adaptive Usage of RTS/CTS

Adaptive usage of RTS/CTS has been recognized as an effective way to deal with hidden nodes in 802.11 networks and it has been used in several rate adaptation schemes such as CARA-like schemes and RRAA. We propose an advanced adaptive RTS scheme in RAM. Similar to the one used in RRAA, our adaptive RTS scheme uses an RTS window (with the size of RTSWnd) to regulate the usage of RTS frames. All data frames within the RTS window shall be transmitted with RTS/CTS support. Moreover, our scheme examines all possible transmission outcomes thoroughly and updates RTSWnd in a timely manner as follows.

Table 2.3 lists two ways of attempting a data frame transmission (i.e., with or without RTS/CTS), possible outcomes of each attempt, and the corresponding actions on updating RTSWnd. Initially, RTSWnd is set to zero to disable RTS usage. The basic heuristic behind our adaptive RTS scheme is that RTSWnd should increase more quickly if a shorter frame is lost (which implies a higher collision

probability), and decrease more quickly if a longer frame succeeds (which implies a lower collision probability). Since an RTS frame is short, an RTS failure indicates that the collision problem may be severe. Hence, we multiply RTSWnd by three. If an RTS transmission succeeds, we decrease RTSWnd slowly (RTSWnd $=$ RTSWnd $-$ 1) regardless whether the subsequent data transmission succeeds or not. This is because an RTS/CTS exchange has already reserved the channel and reduces the probability of collision to the subsequent data transmission. On the other hand, without RTS/CTS support, a successful data transmission (usually with a long transmission duration) indicates a small chance of collision and hence we decrease RTSWnd by half. When a data transmission fails with no preceding RTS, the cause of the failure is unclear. In this situation, we increase RTSWnd slowly (RTSWnd $=$ RTSWnd $+$ 1). We also set a maximum value of 32 for RTSWnd to guarantee stable performance.

Table 2.3 Data Transmission Attempts, Outcomes, and Corresponding Actions on Updating RTSWnd

Data Frame Transmission Attempt	Outcome of Transmission Attempt	Action on Updating RTSWnd
DATA with RTS	RTS Fail	RTSWnd $=3\times$ RTSWnd
	RTS Succ, DATA Fail	RTSWnd $=$ RTSWnd -1
	RTS Succ, DATA Succ	
DATA without RTS	DATA Fail	RTSWnd $=$ RTSWnd $+1$
	DATA Succ	RTSWnd $=$ RTSWnd$/2$

Note that in MadWifi, it is impossible to control the RTS usage on a per-transmission-attempt basis. Therefore, when we implement RAM in MadWifi, the RTS usage is controlled on a per-frame basis. In other words, the transmitter updates RTSWnd when it receives the report from the card firmware after the frame has been transmitted successfully or discarded.

2.4.6 Interoperability between RAM-based and Legacy 802.11 Devices

Given the fact that numerous legacy 802.11 devices have been deployed and will continue to be used in many real-world Wi-Fi applications, it is important to make sure that RAM-based and legacy 802.11 devices can interoperate with each other. We propose an effective scheme in RAM to achieve this goal. Before proceeding to the details of the scheme, we first introduce three modes that a RAM device may operate in: *Legacy mode*, *RAM mode* and *Training mode*. Note that (i) a RAM device maintains a state

transition diagram for each of its communicating partners, meaning that it could operate in different modes to communicate with different partners; (ii) a legacy device always operates in Legacy mode.

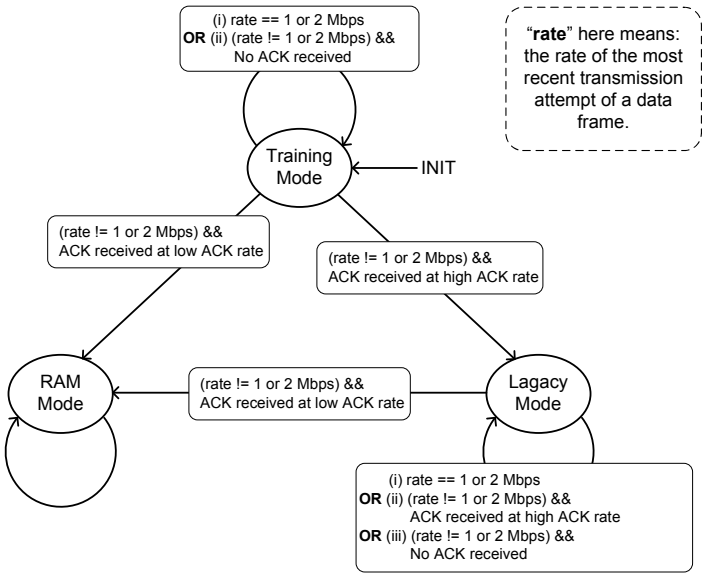
- *Legacy mode*: The transmitter in Legacy mode adopts a non-RAM transmitter-based rate adaptation scheme, e.g., SampleRate. Also, it calculates the NAV value using the default high ACK rate. On the other hand, the receiver in Legacy mode always replies ACK at the default high ACK rate.

- *RAM mode*: The transmitter in RAM mode adopts the mechanism mentioned in Section 2.4.4 to update the transmission rate, and calculates the NAV value using the low ACK rate. The receiver in RAM mode replies ACK at the high or low ACK rate according to the schemes described in Sections 2.4.1, 2.4.2 and 2.4.3.

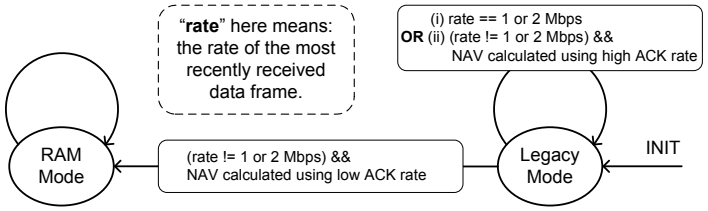
- *Training mode*: Training mode is a transient operation mode and it will switch to RAM or Legacy mode eventually. The transmitter in Training mode adopts a non-RAM transmitter-based rate adaptation scheme, but calculates the NAV value using the low ACK rate.

The state transition diagram for RAM transmitters is shown in Fig. 2.7(a). As shown in the figure, a RAM transmitter always starts at Training mode. In Training mode, the transmitter uses a transmitter-based rate adaptation scheme to update the rate. Meanwhile, it sets the NAV using the low ACK rate to announce that it supports RAM. Note that in Table 2.2 the high and low ACK rates are identical for the data rates of 1 and 2 Mbps. Therefore, the transmitter in Training mode will not switch to other modes if the date rate is 1 or 2 Mbps. Besides, if no ACK frame is received, the transmitter in Training mode also will not trigger a state transition.

In Training mode, when the transmitter receives an ACK frame successfully for a data frame sent at a rate other than 1 and 2 Mbps, a state transition is triggered as follows. If the ACK frame is transmitted at the low ACK rate, the transmitter knows that its communicating partner must be a RAM device and thus switches to RAM mode immediately. On the other hand, if the ACK frame is transmitted at the default high ACK rate, the transmitter considers its communicating partner a legacy device and then switches to Legacy mode, during which it keeps on monitoring the the status of future ACK frames.



(a) State transition diagram for RAM transmitters.



(b) State transition diagram for RAM receivers.

Figure 2.7 State transition diagrams for RAM devices.

Any time it sees an ACK frame transmitted at the low ACK rate for a data frame sent at a rate other than 1 or 2 Mbps, the transmitter switches to RAM mode; otherwise it remains in Legacy mode.

The state transition diagram for RAM receivers is shown in Fig. 2.7(b). As shown in figure, a RAM receiver always starts at Legacy mode. If it receives a data frame with NAV value calculated using the low ACK rate and the data rate is not 1 or 2 Mbps, it realizes that its communicating partner must be a RAM device and then switches to RAM mode immediately. Otherwise, it remains in Legacy mode.

2.5 Experimental Study

In this section, we evaluate the effectiveness of RAM using experimental results. We implement all the RAM modules described in Section 2.4 in MadWifi. We call this complete version of RAM

implementation RAM-FULL. For comparison purpose, we also implement another version of RAM, called RAM-BASIC, which does not include the adaptive RTS module.

2.5.1 Experimental Setup

The hardware and software configurations used in our experiments are listed in Table 2.4. All experiments are conducted using Dell Latitude D620 laptops equipped with CB9-GP-EXT 802.11a/b/g WLAN adaptors, which embed Atheros 5213 chipsets. We use off-the-shelf hardware instead of sophisticated equipments to conduct experiments as this makes our experimental results comparable to what users of ordinary 802.11 devices may expect in realistic scenarios. For outdoor vehicular experiments, we use 7 dBi omni-directional external antennas, which are shown in Fig. 2.8(b).

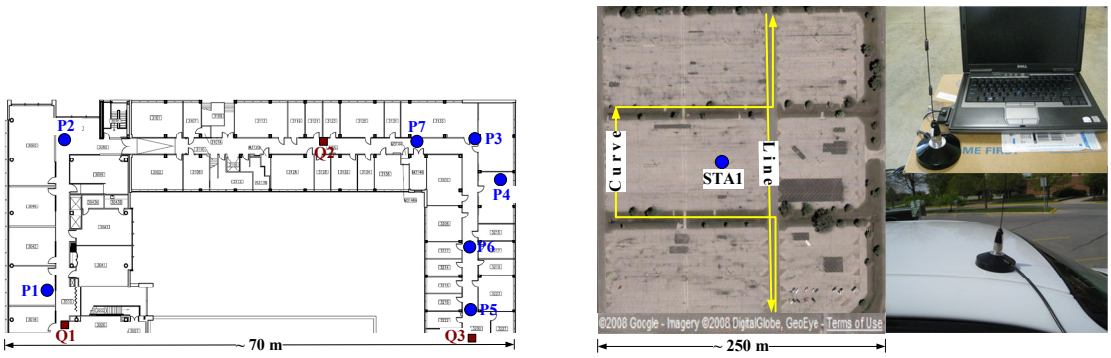
In our experiments, we use Iperf [23] as the UDP packet generator. CBR (Constant Bit Rate) traffic is generated at 30 Mbps with packet size of 1470 octets. The results for each scenario are averaged over five experimental runs. In order to minimize potential unexpected performance variation caused by people's movement and interference from other 802.11 devices, indoor experiments are conducted at nighttime or weekends and outdoor experiments are conducted in the mornings during weekends.

Table 2.4 Configuration Parameters

Parameters	Values
Computer	Dell Latitude D620 Laptop
Operating system	Linux Kernel 2.6.24-16
WLAN adaptor	CB9-GP-EXT 802.11a/b/g
Device driver	MadWifi v0.9.4
802.11 PHY	802.11g
Transmit Power	16 dBm
CBR packet size	1470 octets
CBR rate	30 Mbps

We conduct experiments in both static and mobile scenarios. Indoor experiments (static and mobile) are performed on the 3rd floor of Coover Hall (our department building), as shown in Fig. 2.8(a), and outdoor vehicular experiments are performed in a parking lot near Jack Trice stadium, as shown in Fig. 2.8(b). We mark several locations and moving trajectories on the figures, based on which we design 12 different experimental scenarios, as described in Table 2.5.

We compare the throughput performances of RAM-FULL and RAM-BASIC against the following



(a) Venue for indoor experiments: 3rd floor of Coover Hall.

(b) Venue for outdoor experiments: a parking lot near Jack Trice stadium.

Figure 2.8 Venues for our indoor and outdoor experiments.

Table 2.5 Description of Experimental Scenarios

Scenarios	Descriptions
Static-1	STA1 at P2, STA2 at P1
Static-2	STA1 at P3, STA2 at P5
Static-3	STA1 at P3, STA2 at P2
Static-4	STA1 at P7, STA2 at P6
Walk-1	STA1 at P3, STA2: Q2→Q1→Q2
Walk-2	STA1 at P3, STA2: P3→Q3→P3
Walk-3	STA1 at P4, STA2: P6→Q2→P6
Walk-4	STA1 at P6, STA2: P3→Q2→P3
SlowDrive-1	STA1 is static, STA2 moves along the line up to 25 MPH
SlowDrive-2	STA1 is static, STA2 moves along the curve up to 25 MPH
FastDrive-1	STA1 is static, STA2 moves along the line up to 45 MPH
FastDrive-2	STA1 is static, STA2 moves along the curve up to 45 MPH

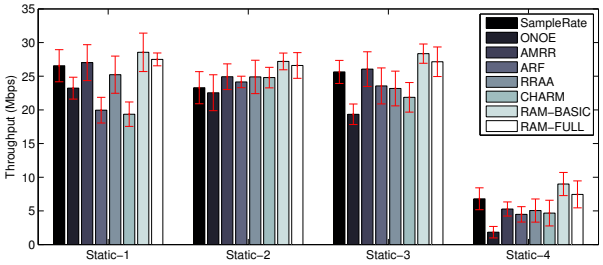
state-of-the-art rate adaptation schemes: SampleRate [7], AMRR [3] ONOE [11], ARF [2], RRAA [6] and CHARM [8]. While SampleRate, AMRR and ONOE have already been implemented in MadWifi, ARF, RRAA and CHARM have not or their codes are not available. Therefore, we implement these three schemes in MadWifi from scratch. Moreover, we do not consider CARA [4] in the performance evaluation because per-transmission-attempt RTS probing and CCA detection proposed in CARA are infeasible in MadWifi.

2.5.2 Experimental Results

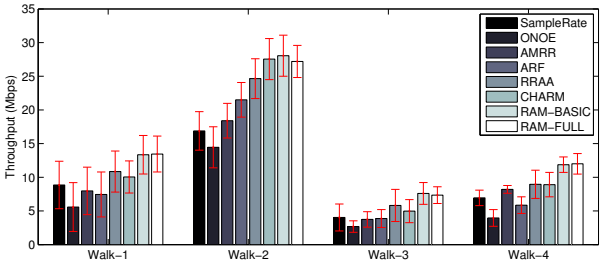
We compare the throughput performances of testing schemes in indoor static, indoor mobile and outdoor vehicular scenarios, and results are plotted in Fig. 2.9. In this section, we first give a few

general observations on the results, and then discuss the results in each of the three scenarios in detail.

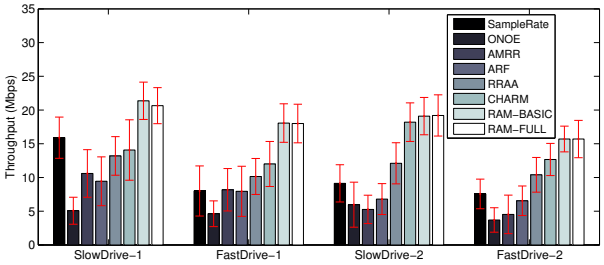
The first general observation is that RAM schemes clearly outperform other testing schemes in all experimental scenarios, indoor or outdoor, static or mobile, and the performance gain becomes more significant as the relative speed of two stations goes up. This is because RAM is receiver-based. By using the feedback from the receiver, the transmitter can select the proper transmission rate to match the current channel condition. In comparison, CHARM is transmitter-based and suffers from channel asymmetry. Moreover, RAM is frame-based, which can adapt much faster to rapid variations of the channel condition. In comparison, SampleRate, AMRR, ONOE and RRAA are transmitter-based schemes and based on packet statistics. As a result, they usually are slow in adapting to the channel variation.



(a) indoor static scenarios.



(b) indoor mobile scenarios.



(c) outdoor vehicular scenarios.

Figure 2.9 Comparison of throughput performances (with 90% confidence interval).

Another general observation is that ARF and ONOE perform the worst in most experimental scenarios. The reason is as follows. ARF waits for 10 consecutive successful transmission attempts before increasing the rate. Unfortunately, from our experiments, we find that channel fluctuation is common in practice, even in indoor static environments. In the presence of channel fluctuation, it is rare to have 10 consecutive successes. In comparison, AMRR – ARF's adaptive variant – adapts the success count threshold by using a binary exponential backoff starting with 10. As a result, AMRR performs better than ARF in scenarios when the channel condition is stable, e.g., indoor static scenarios. However, in mobile scenarios, the performance of AMRR is as poor as ARF.

ONOE is a conservative rate adaptation scheme by design: it increases the transmission rate at most once during any one-second period. Once it decides that a rate does not work well for the current channel condition, it will not attempt this rate again until at least 10 seconds later. From the experiments, we observe that ONOE takes quite long time to converge to the proper transmission rate. This also indicates that the selection of a proper initial transmission rate is critical for ONOE.

2.5.2.1 Indoor Static Scenarios

Experimental results for indoor static scenarios are plotted in Fig. 2.9(a). It can be seen that both RAM-FULL and RAM-BASIC show comparable or better performances than other testing schemes in all four scenarios. RAM-FULL and RAM-BASIC yield similar performances. As will be shown in Sections 2.5.2.2 and 2.5.2.3, their performances are similar as long as there are no hidden nodes in the network. This is because without hidden nodes, the RTS usage is disabled for most of the time and hence RAM-FULL is almost equivalent to RAM-BASIC. For Static-1 scenario, we observe that the performance of CHARM is worse than all others. This is because CHARM is designed based on the assumption of symmetric channel conditions, which does not hold in Static-1 where one station is at P1 (inside an office) and the other station is at P2 (outside the office). From the SNR traces collected in Static-1, we notice that the channel conditions are highly asymmetric and the difference between DATA SNR and ACK SNR is as large as 11 dB.

2.5.2.2 Indoor Mobile Scenarios

Experimental results for indoor mobile scenarios are plotted in Fig. 2.9(b). Similar to indoor static environments, RAM-FULL and RAM-BASIC yield higher throughput than others for all indoor mobile scenarios. Again, because RRAA adopts a time window of 150 ms (hence may not respond to channel variation quickly) and CHARM is unable to handle channel asymmetry (hence may cause unnecessary frame losses or under-utilization of the good channel condition), both RRAA and CHARM perform worse than RAM.

In order to have a good understanding on how and why RAM schemes outperform others, we investigate the Walk-1 scenario in more depth and study the cause of the observed throughput differences by plotting the rate usage distribution of each scheme in Fig. 2.10. The numbers of successful or failed transmission attempts at different transmission rates are shown as positive or negative bars of different colors. As shown in the figure, RAM schemes make effective usage of the available transmission rates: (i) a majority of the successfully transmitted frames are attempted at the highest rates of 48 or 54 Mbps; (ii) a very few frames are transmitted at the lowest rates of 1 or 2 Mbps; and (iii) the frame loss ratio is low (3.63% for RAM-FULL and 3.87% for RAM-BASIC). In comparison, both RRAA and CHARM suffer a much higher frame loss ratio at 10.56% and 12.70% respectively. Moreover, we observe that SampleRate and ONOE transmit a large portion of the frames at low rates due to their conservative natures.

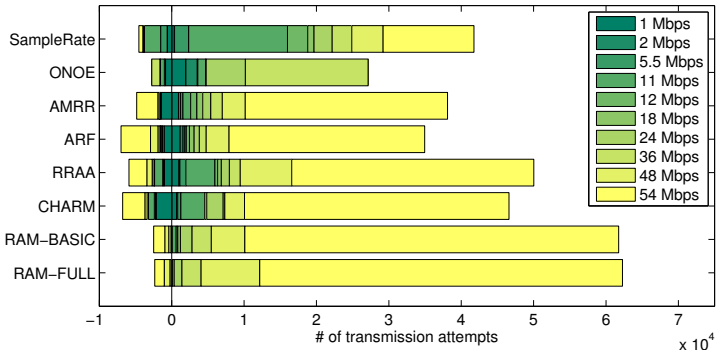
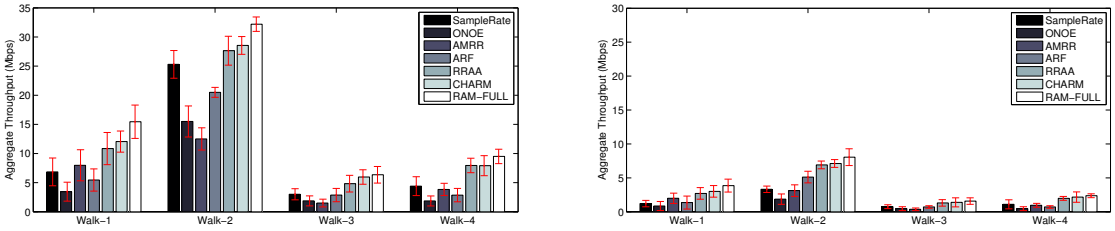


Figure 2.10 Rate usage distribution for Walk-1. The number of successful or failed transmission attempts is shown as a positive or negative bar.

To evaluate the performance of RAM further, we have done a few additional experiments under

more complicated indoor mobile scenarios: (i) *duplex communication*, where two stations transmit data to each other simultaneously; and (ii) *interfering communication*, where two additional communication pairs are introduced as the interfering sources to operate on the same channel as the target communication pair that we evaluate. In (ii), one of the interfering pairs are co-located at the P4 position shown in Fig. 2.8(a) and communicate at high rates, while the other pair are placed at the P2 and P3 positions and communicate at low rates. Results are plotted in Fig. 2.11, where the superior performance of RAM-FULL over other schemes can be observed.



(a) indoor mobile scenarios with duplex communication.

(b) indoor mobile scenarios with interfering communication.

Figure 2.11 Comparison of throughput performances in more complicated indoor mobile scenarios (each point is plotted with 90% confidence interval).

2.5.2.3 Outdoor Vehicular Scenarios

Fig. 2.9(c) shows the experimental results for outdoor vehicular scenarios and Fig. 2.12 plots the rate usage distribution of each testing scheme in the FastDrive-1 scenario. In general, the results are similar to those in indoor mobile environments. In addition to the similar observations discussed in the previous section, we have a few more observations as follows.

Firstly, from the SNR traces collected (not included in the paper due to space limitation), we observe that the channel condition in outdoor vehicular environments (with faster station movement) fluctuates more frequently and at a larger scale than indoor mobile environments (with slower station movement).

Secondly, in spite of the high fluctuation of channel conditions in outdoor vehicular environments, RAM continues to outperform all other schemes and the performance gain becomes more significant. As shown in Fig. 2.12, when using RAM-FULL in FastDrive-1, 81.27% of the frames are transmitted successfully at 48 or 54 Mbps while the frame loss ratio is only 3.15%.

Thirdly, as shown in Fig. 2.12, both RRAA and CHARM suffer an even high frame loss ratio (13.55% and 14.08% respectively for FastDrive-1, in comparison to 10.56% and 12.70% for Walk-1) due to more dynamic channel conditions in outdoor vehicular environments.

Lastly, comparing Fig. 2.12 with Fig. 2.10, we can see that ONOE behaves quite differently in FastDrive-1 and Walk-1 scenarios. In FastDrive-1, ONOE transmits frames at much higher rates for most of the time but the frame loss ratio is also higher. This is because we use different initial transmission rates for ONOE in FastDrive-1 and Walk-1 scenarios.

2.5.3 Interoperability between RAM-based and Legacy 802.11 Devices

As discussed in Section 2.4.6, it is important that RAM-based devices can interoperate with legacy 802.11 devices. In this section, we verify the effectiveness of our proposed solution to interoperability using experimental results. We design four experimental scenarios which are listed in Table 2.6. There are two communication pairs in the experiments. Communication pair 2 uses legacy 802.11 devices running the SampleRate rate adaptation scheme, while communication pair 1 uses different combinations of RAM-FULL-based and legacy 802.11 devices in different scenarios.

In the first experiment, we keep the transmitter of communcation pair 2 static at the P3 location shown in Fig. 2.8(a). Results are shown in Fig. 2.13(a). When communication pair 1 proceeds independently, it achieves a throughput of 15.2 Mbps when both stations use RAM-FULL. On the other hand, if communication pair 2 proceeds independently, it achieves a throughput of 25.6 Mbps. As we can see from Fig. 2.13(a), the performance of each link decreases when both links are active and

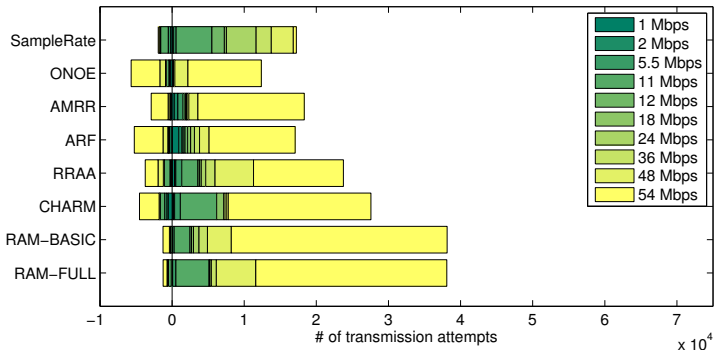
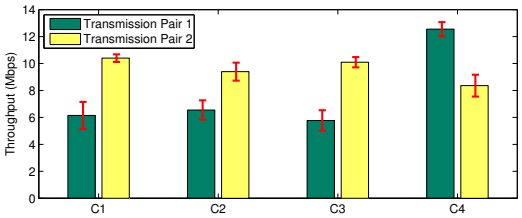


Figure 2.12 Rate usage distribution for FastDrive-1. The number of successful or failed transmission attempts is shown as a positive or negative bar.

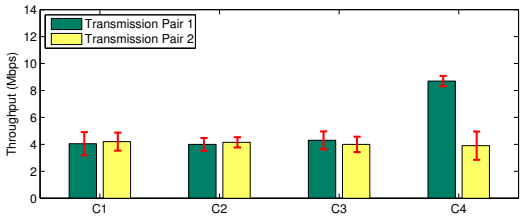
Table 2.6 Description of Experimental Scenarios for Interoperability between RAM-based and Legacy 802.11 Devices

Scenarios	Communication Pair 1		Communication Pair 2	
	Tx (P6→Q2→P6)	Rx (at P4)	Tx (various locations)	Rx (at P4)
C1	Legacy	Legacy	Legacy	Legacy
C2	Legacy	RAM-FULL	Legacy	Legacy
C3	RAM-FULL	Legacy	Legacy	Legacy
C4	RAM-FULL	RAM-FULL	Legacy	Legacy

contend for the channel.



(a) Communication pair 1 is mobile while communication pair 2 is static. The transmitter of communication pair 2 is static at P3.



(b) Both communication pairs are mobile. The transmitter of communication pair 2 moves in the same trajectory of P6→Q2→P6 as the transmitter of communication pair 1.

Figure 2.13 Experimental results for interoperability between RAM-based and legacy 802.11 devices (each point is plotted with 90% confidence interval).

As discussed in Section 2.4.6, by using our proposed solution, receiver of communication pair 1 in scenario C2 and transmitter of communication pair 1 in C3 will eventually operate in Legacy mode. It means that C1, C2 and C3 shall have a similar throughout performance, which can clearly be observed in Fig. 2.13(a). On the other hand, stations of communication pair 1 in C4 will operate in RAM mode eventually, which yields a much higher throughput than other scenarios. Moreover, we observe that the performance of communication pair 2 in C4 actually decreases a bit comparing to other scenarios. This is because RAM calculates the NAV value in a conservative manner, which may give more chance for RAM-based devices to contend for the channel when ACK is sent at the high rate. This issue could be addressed by introducing another timer and assigning it a value equal to the NAV value; RAM-based device will not attempt to access the channel until this timer expires, thus contending fairly with other devices in the network.

In the second experiment, the transmitter of communication pair 2 moves in the same trajectory as

the transmitter of communication pair 1. Results are shown in Fig. 2.13(b) and similar observations can be made as in the first experiment.

2.6 Simulation Study

In this section, we use the ns-2 simulator [24] to evaluate the effectiveness of RAM schemes in the presence of hidden nodes, as well as the effects of ACK rate variation on the system performance. The reasons for using simulation instead of experiment to study these issues are as follows: (i) in practice, it is difficult to set up a hidden nodes scenario; and (ii) to study the effects of ACK rate variation, we need to compare RAM with an ideal rate adaptation scheme that can only be achieved in the simulation. In the following, we first introduce the simulation setup, followed by simulation results.

2.6.1 Simulation Setup

Instead of using the simple 0/1 packet delivery model given in the ns-2 simulator, we use the empirical PDR (Packet Delivery Rate) vs. SNR curves in the simulations. These curves (shown in Fig. 2.14) are obtained from 10 experimental traces with each lasting for about 10 minutes. Moreover, for the clarity of presentation and explanation, we assume the 802.11b PHY in the simulations. Simulations for other 802.11 PHYs yield similar results.

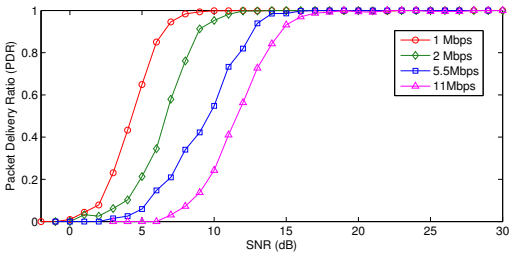


Figure 2.14 The PDR vs. SNR curves for 802.11b rates used in the simulation.

In the simulations, each transmitter transmits in the saturation mode, i.e., its data queue is never empty, and all data frames are transmitted without fragmentation. We use LLC/IP/UDP as the upper layer protocol suite, and the MAC-layer data payload length is 1500 octets.

2.6.2 Simulation Results for Hidden Nodes Scenario

In the first part of the simulation, we investigate the performances of testing schemes in the presence of hidden nodes. The simulated hidden nodes scenario is shown in Fig. 2.15, where two transmitters are located at opposite sides of the receiver. Both transmitters start moving at 5 m/s at the same time and they are hidden to each other along most of the trajectory. We simulate a Ricean fading channel with a K-factor of 6 dB and assume a maximum speed of 10 m/s for movement in the environment.

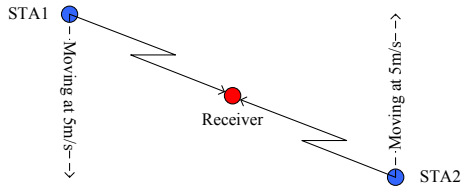
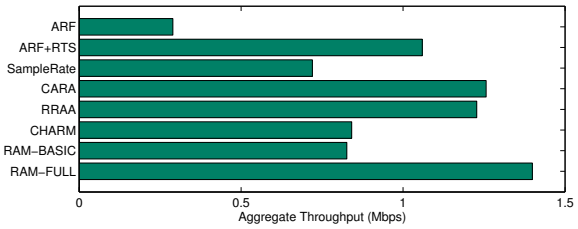


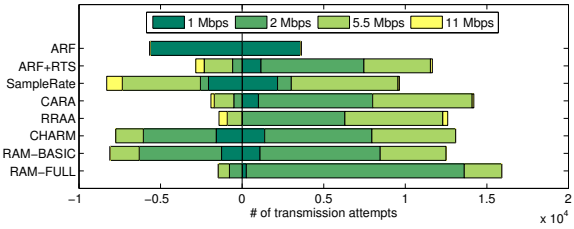
Figure 2.15 The simulated hidden nodes scenario.

We compare the throughput performances of RAM schemes against the following rate adaptation schemes: SampleRate, ARF, ARF+RTS (ARF with RTS enabled all the time), RRAA, CHARM and CARA. Note that CARA is a rate adaptation scheme that is not implementable with commercial 802.11 devices and hence is not evaluated in our experimental study.

Throughputs of testing schemes are compared in Fig. 2.16(a). As expected, schemes with RTS capabilities such as RAM-FULL, ARF+RTS, CARA and RRAA can deal with hidden nodes well and yield higher throughput. Among them, RAM-FULL has the best performance. In comparison, schemes without RTS capabilities suffer significant performance degradation, including RAM-BASIC, CHARM, ARF and SampleRate. ARF performs particularly bad because it cannot differentiate collision-induced losses from channel-error-induced losses and hence transmits at a very low rate. Note



(a) Throughput comparison.



(b) Comparison of rate usage distributions.

Figure 2.16 Simulation results for the hidden nodes scenario.

that CHARM yields comparable throughput as RAM-BASIC. This is because we use the Ricean fading model to simulate a perfect symmetric channel, and hence CHARM does not suffer from asymmetric channel conditions.

We plot the rate usage distributions of testing schemes in Fig. 2.16(b). High frame loss rate caused by hidden nodes can be seen in the figure for RAM-BASIC, CHARM, ARF and SampleRate. Among them, ARF has the highest frame loss ratio of 61%. In comparison, when RTS is used to deal with hidden nodes, frame loss ratio is reduced drastically for RAM-FULL, ARF+RTS, CARA and RRAA. RAM-FULL has the largest number of successful transmission attempts and the smallest frame loss ratio, which explains its best throughput performance.

2.6.3 Effects of ACK Rate Variation on the System Performance

In the second part of the simulation, we study the effects of ACK rate variation used in RAM schemes on the system performance. In RAM schemes, since we set the duration field using the low ACK rate, when the receiver replies ACK at the high rate, the channel will be idle for a short period of time. To verify that this mechanism will not cause noticeable performance degradation, we simulate multiple Tx-Rx pairs transmitting simultaneously and stations are randomly placed within a circle with a radius of 40 m. Transmitters always transmit in the saturation mode.

Table 2.7 Throughput Comparison (in Mbps) with Multiple Simultaneous Tx-Rx Pairs

# Tx-Rx Pairs	1	2	4	8	16	32
Ideal-RAM	4.5828	4.0640	3.9815	3.8752	3.1668	2.1563
RAM-FULL	4.5822	3.9989	3.9555	3.7992	3.0887	2.0291

We compare RAM-FULL with a scheme called Ideal-RAM in terms of the system throughput. Ideal-RAM operates in the same way as RAM-FULL except that Ideal-RAM does not use the ACK rate variation to convey the feedback information; rather, the rate selections by the receiver are made available to the transmitter by modifying the ns-2 simulator. So Ideal-RAM is only possible with the simulator but not implementable in practice. We vary the number of Tx-Rx pairs and simulation results (averaged over 20 simulation runs) are given in Table 2.7. We can see that even when the network is heavily loaded with 32 Tx-Rx pairs, the ACK rate variation used in RAM-FULL only results in a small 5.9% degradation of the system throughput.

2.7 Conclusions

From the experiments, we find that conditions of wireless channels in mobile environments exhibit severe asymmetry and high fluctuation. To deal with these issues, we propose a practical rate adaptation scheme, called RAM, for 802.11 devices in mobile environments. RAM is a receiver-based scheme and the main novelty of RAM lies in the usage of ACK rate variation to convey the feedback information from the receiver to the transmitter. Moreover, a RAM-based 802.11 device is able to identify whether its communicating partner is a RAM-based or legacy 802.11 device, based on which it operates in the corresponding mode. This way, the interoperability between RAM-based and legacy 802.11 devices can be guaranteed.

We have implemented RAM in the MadWifi device driver and extensive experimental results show that RAM is able to deal with channel symmetry and adapts quickly to the channel variation. It outperforms existing rate adaptation schemes in both static and mobile environments, particularly in outdoor vehicular scenarios.

CHAPTER 3. FAST AND SEAMLESS HANDOFF WITH NULL DWELL TIME

3.1 Introduction

Nowadays, more and more large-scale 802.11 networks with multiple APs have been deployed in public venues and even city-wide in order to provide broadband wireless networking services to a larger area since it has be considered as an effective and economic way to improve the capacity and performance of 802.11 networks. One of the important goals of deploying such networks is to support seamless data roaming when users are mobile. For example, a user making VoIP (Voice over IP) calls from an 802.11 enabled smart phone should not experience service disruption or quality degradation to the calls when the user moves around in the network. In practice, this is a challenging task due mainly to the long delay incurred in the 802.11 handoff procedure, during which the data services are disrupted.

One of the core components in the 802.11 handoff procedure is to collect the information about other channels and nearby APs operating on these channels. In a typical 802.11 handoff procedure, a station switches to a new channel, sends a probe frame, and then dwells on the channel for a certain amount of time waiting for responses from the APs (if any) operating on the channel. Channel dwell time has been recognized as the most significant contributor to the overall handoff delay. In general, it is desirable to minimize channel dwell time as much as possible to reduce the handoff delay and minimize the disruption to data roaming services. On the other hand, the station needs to dwell on the scanned channel long enough to wait for returning response frames. If the APs on the channel are busy serving other stations, or if there are many APs operating on the channel, the station may not be able to collect all the responses when channel dwell time is set too small. So there is a tradeoff. In practice, it is difficult to set a proper channel dwell time for an 802.11 station since the context information about the scanned channel, such as the number of APs on the channel and the loads of the APs on the channel,

is unavailable to the station. Various fast handoff schemes have been proposed in the past. However, they either use simple heuristics to set a small channel dwell time, which may not perform well under certain circumstances, or require modification or upgrading of the already deployed AP infrastructure or the 802.11 device firmware.

3.2 Literature Survey

Various fast handoff schemes have been proposed in the past. Most of them have focused on reducing channel dwell time. In [25], the authors showed that the scan phase is the most significant contributor to the overall handoff delay and the variations in channel dwell time account for the large variations in the handoff delay. The schemes proposed in [26, 27] try to reduce the number of channels to probe during the scan phase. For example, [27] adopts a neighbor graph approach and the station only probes the channels that have active APs operating on. D-Scan [28], Proactive-Scan [29], [27], and [30] try to reduce channel dwell time by tuning the values of `MinChannelTime` and `MaxChannelTime` in the case of active scan. SyncScan [31] reduces the probe duration by having all APs synchronize their Beacon frames in the case of passive scan. D-Scan, Proactive-Scan and the scheme in [32] interleave the long scan phase to reduce the packet delay. Different from the above schemes which all try to reduce the scan delay, the scheme in [33] aims to reduce the re-association delay.

During the handoff procedure, another important issue is for an 802.11 station to decide whether to re-associate with a new AP. One of the commonly-used metrics is the signal strength of an AP. For example, hysteresis-based approaches in [28, 29, 32] direct a station to re-associate with the new AP that has a stronger signal strength (by a certain threshold) than its current AP. [26] compares several re-association metrics that are based on signal strength statistics. [34] combines the signal strength value with the large-time-scale performance measurement. In [35], another metric is proposed which is called available capacity. It is the product of the percentage of free air time and the expected transmission rate.

3.3 Observations & Motivations

In this section, we present a few interesting observations from experiments which help us understand the tradeoffs involved in the handoff procedure and design our proposed scheme. Experiments

are conducted with Dell Latitude laptops equipped with D-link WNA-1330 802.11b/g and Wistron CB9-GP-EXT 802.11a/b/g cards. Each laptop is loaded with the MadWifi device driver v0.9.4 [20] to collect experimental data.

3.3.1 Effects of Channel Dwell Time and AP Load

In general, it is desirable to minimize channel dwell time as much as possible to reduce the handoff delay and minimize the disruption to data services. On the other hand, the station needs to dwell on the scanned channel long enough to wait for returning Probe Response frames. If the APs on the channel are busy serving other stations, or if there are many APs on the channel, the station may not be able to collect all the Probe Responses when channel dwell time is set too small. From the experimental results plotted in Fig. 3.1, it can be seen clearly that the percentage of Probe Response frames received during channel dwell time drops significantly as channel dwell time reduces or as the AP load goes up. So there is a tradeoff. In practice, it is difficult to set a proper channel dwell time for an 802.11 station since the context information about the scanned channel, such as the number of APs on the channel and the loads of the APs on the channel, is unavailable to the station.

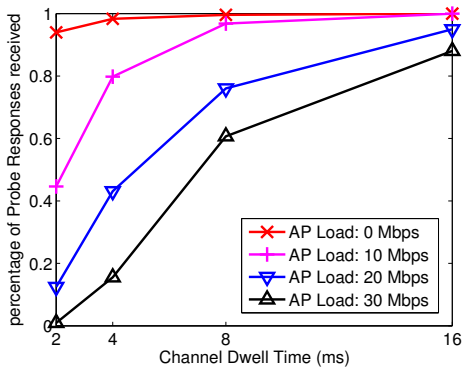


Figure 3.1 Effects of channel dwell time and AP load on the percentage of Probe Response frames received

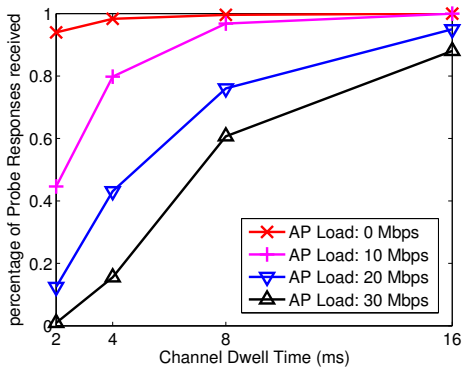


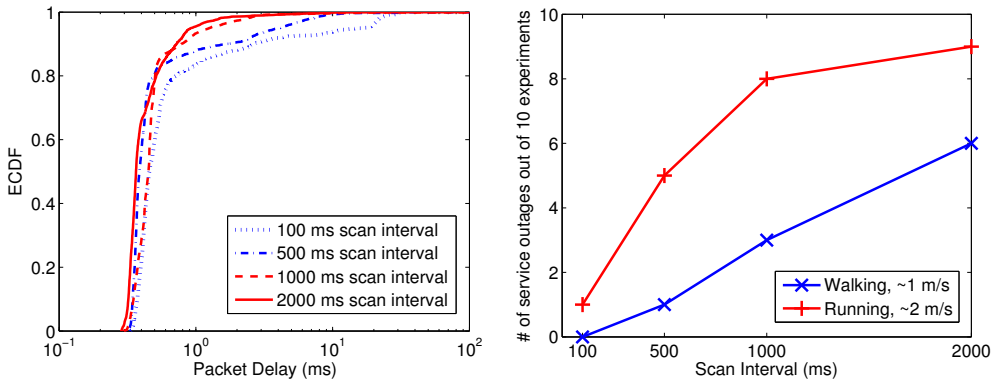
Figure 3.2 Effects of probe interleaving on packet delay. Scan intensity is 1 channel per 100 ms. Channel dwell time is set to 20 ms.

3.3.2 Effects of Probe Interleaving

We define *scan intensity* as the number of channels to probe per unit time. To maintain the same scan intensity level, an 802.11 station has two options: (i) probe all channels during one single scan phase with a long scan interval; or (ii) probe fewer channels during one scan phase but with a shorter scan interval. We call the latter one the *probe interleaving* option. Since data transmissions are disrupted during the scan phase when the station switches to other channels, we expect a better packet delay performance when channel probing is more interleaved, which is confirmed by the experimental results plotted in Fig. 3.2. As shown in the figure, in the case of 1 probe per scan phase (with a scan interval of 100 ms), the maximum packet delay is around 30 ms, while in the case of 11 probes per scan phase (with a scan interval of 1100 ms), the maximum packet delay is around 295 ms, which is unacceptable for real-time data transmissions. This observation is also consistent with the observations from [29, 32].

3.3.3 Effects of Scan Intensity

We now study the effects of scan intensity on the handoff performance by varying the scan interval from 100 ms to 2000 ms with a fixed number (i.e., one) of channels to probe per scan phase. The first experiment is performed between an AP and a static station. Fig. 3.3(a) plots the relation between scan intensity and packet delay. As shown in the figure, packet delay decreases as the scan intensity level decreases (i.e., as the scan interval increases). In the second experiment, we set up two APs with overlapping coverage and have a mobile station moving from one AP to another at different speeds (walking and running). The station hands off to the new AP when the signal strength of the new AP is 5 dB higher than that of its current AP. We define *service outage* as the event when the station loses the connection to the current AP but has not associated with the new AP yet. Fig. 3.3(b) plots the relation between scan intensity and number of service outages. As shown in the figure, as the scan intensity level decreases, the probability for an 802.11 station to experience a service outage increases. So there is a tradeoff and it is a challenging task to set a proper scan intensity level in practice.



(a) packet delay vs. scan intensity (b) service outage vs. scan intensity

Figure 3.3 Effects of scan intensity. We fix the number of channels to probe per scan phase to one, and vary the scan intensity level by varying the scan interval. Channel dwell time is 20 ms.

3.3.4 Potential Performance Anomaly Caused by Handoff

As discussed in Section 3.2, another important issue during the handoff procedure is to decide whether an 802.11 station should re-associate with a new AP, and most existing schemes make the decision based on the comparison of the signal strengths of the APs. We conduct a simple experiment to study the performance of such simple *strongest-signal-first* heuristic.

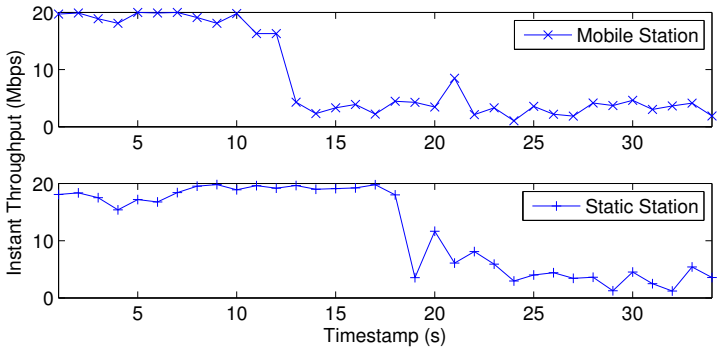


Figure 3.4 Performance anomaly caused by handoff.

In the experiment, a static station associates with AP_1 and transmits at a high rate of 54 Mbps. A mobile station moves from AP_2 towards AP_1 and re-associates with AP_1 at around the 18-second mark when the signal strength of AP_1 is higher than that of AP_2. However, at that moment, the mobile station is still at quite a distance away from AP_1 and hence communicates with the AP at a low rate. Fig. 3.4 plots the instant throughputs of the two stations, where *instant throughput* is defined as the

throughput measured for the one-second period ending at the corresponding time instance. We can see that the throughput of the static station drops sharply right after the mobile station joins the network. This, in fact, is the well-known *performance anomaly* in 802.11 networks [36], which is caused by the transmission rate diversity among all stations associating with the same AP. The high-rate station is "slowed down" by the low-rate station because the 802.11 protocol is designed to allow all contending stations to access the channel with equal probability. Various schemes have been proposed to deal with this issue. Unfortunately, during the conventional 802.11 handoff procedure, since the context information about the APs such as the transmission rates of their associated stations is unavailable to the mobile station, such performance anomaly is difficult to avoid.

3.4 Design and Implementation of HaND

In this section, we first introduce the overall structure of the proposed HaND scheme, and then describe the station and AP behaviors in detail.

3.4.1 Overall Structure of HaND

The overall structure of HaND is shown in Fig 3.5. Suppose that the station currently associates with AP_{curr} which operates on channel CH_{curr}, and another access point AP_{new} operates on channel CH_{new}. HaND works in the following steps:

(i): The station switches to channel CH_{new} and broadcasts a Probe Request frame, and then switches back to its working channel, i.e., CH_{curr}, immediately without dwelling on channel CH_{new}.

(ii): Upon receiving the Probe Request frame from the station, AP_{new} immediately replies a Probe Response frame via its wireless interface `ath0`. Note that since the station has already switched back to channel CH_{curr}, it will not be able to receive this Probe Response frame. However, the presence of this step is important since it guarantees the backward compatibility of the proposed HaND scheme with other 802.11 stations that follow the default handoff procedure. More details will be discussed in Section 3.4.2.

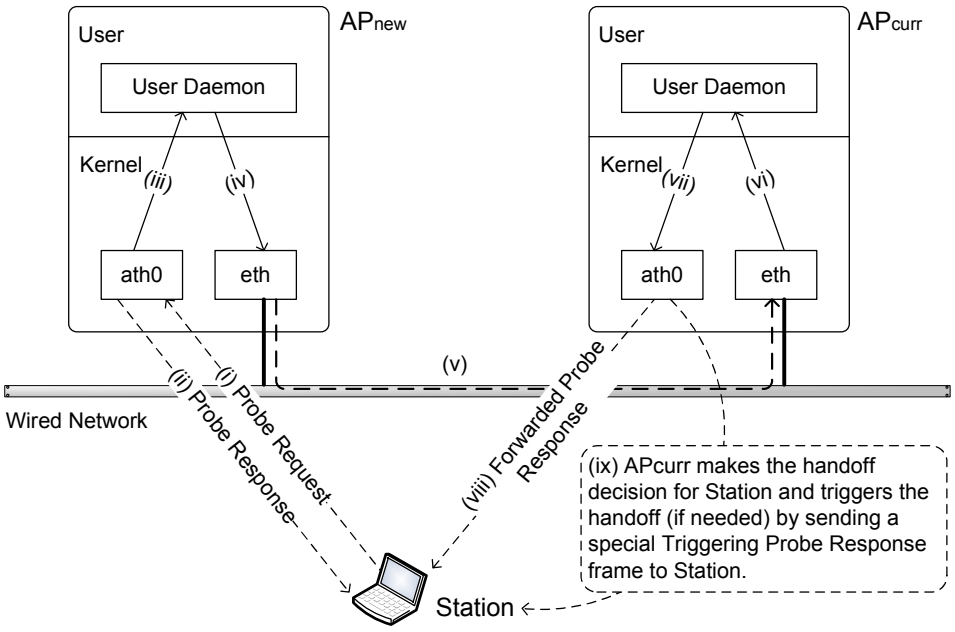


Figure 3.5 Overall structure of the proposed HaND scheme.

(iii): In addition, AP_{new} passes the information about its own channel (i.e., CH_{new}), the RSSI value of the received Probe Request frame, and a copy of the generated Probe Response frame to a user daemon running in the user space.

(iv,v,vi): The user daemon at AP_{new} in turn forwards these information to a user daemon at AP_{curr} via the wired interface `eth` and the wired communication backbone between APs.

(vii,viii): Upon receiving the forwarded information, AP_{curr} updates a local database that stores all the information forwarded from neighboring APs, generates a special *Forwarded Probe Response* frame, and broadcasts it via its wireless interface `ath0` to notify the station of the information about AP_{new}. This way, the station gets to collect neighboring APs' information without wasting time to dwell on other channels. Details will be discussed in Section 3.4.3.3.

(ix): A unique characteristic of HaND is that the AP_{curr} makes the decision for the station on whether to re-associate with AP_{new}, instead of the station itself making the decision. This becomes possible with HaND because AP_{curr} coordinates with neighboring APs and collects their information via the wired communication backbone. The actual handoff is triggered by a special *Triggering Probe Response* frame sent by AP_{curr} to the station. Details will be discussed in Section 3.4.3.4.

3.4.2 HaND Station Behaviors

Depending on whether it currently associates with an AP, a HaND station behaves in two different ways. If it currently does not associate with any AP, it performs a regular scan (in comparison to the adaptive scan discussed below) immediately. During the regular scan, the station probes all channels and channel dwell time for each channel is 20 ms. After collecting all the information about nearby APs on all channels, it associates with the AP that has the highest RSSI value. On the other hand, if a HaND station currently associates with an AP (denoted as AP_{curr}), its behaviors become more complicated, which we describe from the following three aspects: *when to trigger a scan*, *how to scan* and *which channel to scan*.

3.4.2.1 When to Scan

The station maintains a moving average of the RSSI values collected from downlink traffics from AP_{curr} to the station such as data transmissions and Beacon broadcasts, and another moving average of uplink RSSI values based on the information carried in the Forwarded Probe Response frames. To deal with potential link asymmetry between the station and AP_{curr}, we use the minimum of these two moving averages (denoted as $RSSI_{AP_{curr}}$) to reflect the link quality between them. When $RSSI_{AP_{curr}}$ is below a certain threshold (35 dB in our implementation. In MadWifi, the reported RSSI value is indeed the SNR (Signal-to-Noise Ratio) value and hence the unit is dB [21].), the station starts scanning other channels. Neighboring APs' information such as the RSSI values will be reported to the station by AP_{curr} via Forwarded Probe Request frames. This procedure will be described in detail in Section 3.4.3.3. Reported information will be used to decide *how to scan* and *which channel to scan*.

3.4.2.2 How to Scan

An important parameter to characterize a station's scan behaviors is the *scan interval* (denoted as I_s). In HaND, the value of I_s varies with $RSSI_{AP_{curr}}$. In general, when the link quality with AP_{curr} gets worse ($RSSI_{AP_{curr}}\downarrow$), the station should increase the scanning frequency ($I_s\downarrow$) to search for potential backup APs to re-associate with. In our implementation, we set three values for I_s: 450 ms if $RSSI_{AP_{curr}} \in$ [25,35) dB, 300 ms if $RSSI_{AP_{curr}} \in$ [15,25) dB and 150 ms if $RSSI_{AP_{curr}}$ is below 15 dB.

Since data services are disrupted during the scan phase, HaND does the following to minimize the effects: (i) only one of the channels is probed every I_s time; and (ii) the station switches to the channel to send a Probe Request frame and then switches back to its current working channel immediately with zero channel dwell time. This achieves the purpose because:

- Channel switching time typically is very small, e.g., around 4.8 ms for cards with Atheros chipsets, based on our experiments, and around 2.9 ms for cards with Intel chipsets, according to [29];
- Transmission time of the Probe Request frame is negligible since it has a maximum size of 72 octets [1] which can be transmitted in less than 1 ms even at the lowest rate of 1 Mbps.

3.4.2.3 Which Channel to Scan

During each scan phase, the station selects a channel to probe on a probabilistic basis. In general, channels with operating APs that have a good link quality to the station are scanned with a higher probability. It works as follows. Let m denote the total number of channels. Then, based on the information collected from downlink traffics from AP_{curr} as well as from Forwarded Probe Response frames, the station maintains an AP RSSI vector:

$$\overrightarrow{\text{RSSI}_{\text{AP}}} = \{\text{RSSI}_{\text{AP}_{1,1}}, \cdots, \text{RSSI}_{\text{AP}_{2,1}}, \cdots\cdots, \text{RSSI}_{\text{AP}_{m,1}}, \cdots\},$$

where $\text{RSSI}_{\text{AP}_{i,j}}$ is the moving average of the RSSI values for the j-th AP operating on channel CH_i, where $1 \leqslant i \leqslant m$. Then, as shown in Fig. 3.6, the station maps the AP RSSI vector to a channel RSSI vector:

$$\overrightarrow{\text{RSSI}_{\text{CH}}} = \{\text{RSSI}_{\text{CH}_1}, \text{RSSI}_{\text{CH}_2}, \cdots, \text{RSSI}_{\text{CH}_m}\},$$

where

$$\text{RSSI}_{\text{CH}_i} = \max_j \text{RSSI}_{\text{AP}_{i,j}}, \;\; 1 \leqslant i \leqslant m.$$

Note that if no AP operates on a channel, the corresponding RSSI_{CH} is set to minus infinity. Then, the channels are sorted into two groups: G_{H} and G_{L}, where

$$G_{\text{H}} = \Big\{\text{all CH}_i \text{ with } \text{RSSI}_{\text{CH}_i} \geqslant \min\{\text{TH}_{\text{RSSI}}, \text{RSSI}_{\text{AP}_{\text{curr}}}\}\Big\};$$
$$G_{\text{L}} = \Big\{\text{all CH}_i \text{ with } \text{RSSI}_{\text{CH}_i} < \min\{\text{TH}_{\text{RSSI}}, \text{RSSI}_{\text{AP}_{\text{curr}}}\}\Big\}.$$

The threshold TH_{RSSI} is a design parameter and it is set to 15 dB in HaND. When the station chooses a channel to probe, it first chooses between G_{H} (with a probability of P_{H}) and G_{L} (with a probability of P_{L}). In our implementation, P_{H} and P_{L} are set to 0.9 and 0.1, respectively. Then, one of the channels from the chosen group is randomly selected (with a uniform distribution) to be probed.

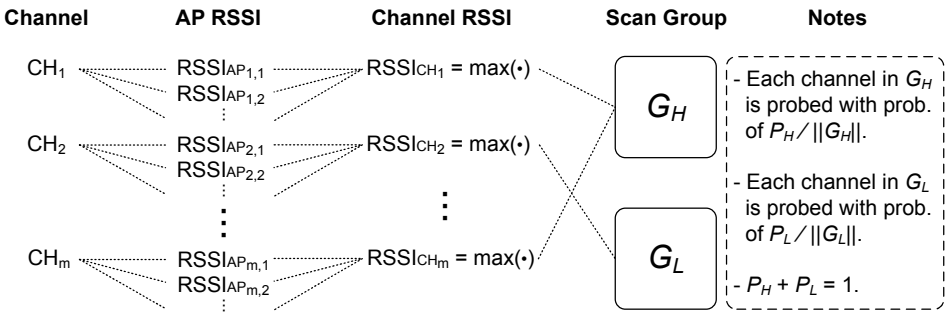


Figure 3.6 HaND station selects a channel to probe on a probabilistic basis.

3.4.3 HaND AP Behaviors

The main innovations of the proposed HaND scheme reside on the AP side. In this section, we describe the behaviors of a HaND AP from the following five aspects.

3.4.3.1 Probe Request Overhearing Avoidance

From the experiments, we notice that an AP may overhear the Probe Request frames transmitted on other channels different from the one it operates on, due to imperfect signal filter implementation in practical 802.11 devices. Clearly, the AP should not respond to these requests since the RSSI measurement is skewed and does not reflect the actual link quality between the AP and the station that sends the probe request.

To filter out the overheard Probe Request frames, we propose to include in each Probe Request frame the index of the channel that it is intended to probe. Then, upon receiving a Probe Request frame, the AP checks the channel index. If it matches the AP's own operating channel, the request is accepted; otherwise it is discarded. This proposal is practically feasible and can be implemented with commodity 802.11 devices since the 802.11 management frames (e.g., Probe Request, Probe Response, etc.) usually are generated in the device driver, which is different from the 802.11 control frames (e.g., RTS, CTS and ACK) whose generation process typically is hard-coded in the device firmware. In the HaND implementation, we modify the content of *SSID Information Element (IE)* in the Probe Request frame for this purpose.

3.4.3.2 Information Exchange over the Wired Network

The key idea of the proposed HaND scheme is to leverage on the wired communication backbone between APs to relay the context information about wireless channels.

As shown in Fig. 3.5, upon receiving a Probe Request frame, AP_{new} takes two actions. Firstly, it replies a Probe Response frame immediately via its wireless interface. This is to make sure that the HaND AP is backward-compatible with regular 802.11 stations that run the default handoff procedure. Secondly, it passes the following information to a user daemon: the index of the channel it operates on, the RSSI value of the received Probe Request frame, and a copy of the generated Probe Response frame. Then, the user daemon updates a local table where each entry of the table corresponds to an 802.11 station in the network and it contains:

- MAC_{AP}: the MAC address of each AP it has probed;
- $RSSI_{AP}$: the moving average of the RSSI values of the Probe Request frames received by each AP it has probed (including the current AP);
- Contents of the most recent Probe Response frame generated by each AP the station has probed.

Any time when a table entry is updated, it will be sent to other APs. This is the first piece of the context information exchanged over the wired network. Another piece of context information exchanged between APs includes their own IP addresses as well as transmission rates and MAC addresses of their

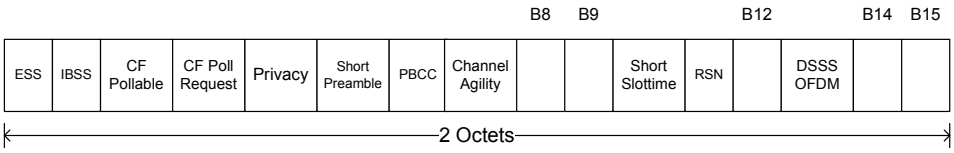


Figure 3.7 Format of the Capability Information fixed field. There currently are five reserved bits: B8, B9, B12, B14 and B15. In HaND, we use B9, B12, B14 and B15 (called `CHAN_BITS`) to carry the channel index and use B8 (called `TRIG_BIT`) to specify a Triggering Probe Response frame.

associated stations. This information is exchanged periodically and used by an AP to make the handoff decisions for its associated stations. Details will be discussed in Section 3.4.3.4.

3.4.3.3 Forwarded Probe Response Frames

After receiving the updated table entry from AP_{new}, AP_{curr} updates its local table accordingly and sends a *Forwarded Probe Response* frame to the corresponding station. Forwarded Probe Response frame is a special Probe Response frame with modified frame fields. To convey the information about AP_{new}, the following fields are modified in the Forwarded Probe Response frame: *source address* (set to the MAC address of AP_{new}), *the Length field of the DS Parameter Set IE* and *the reserved bits in the Capability Information fixed field*. Note that these fields are not used or checked by the legacy 802.11 devices; as a result, our proposed scheme will not cause any confusion or compatibility issue to the legacy 802.11 devices.

According to the IEEE 802.11 standard, a Probe Response frame contains several IEs and fixed fields, including *DS Parameter Set IE* and *Capability Information fixed field*. In the current 802.11 implementation, the eight-bit `Length` field of the DS Parameter Set IE has a constant value of one and is not used or checked in Madiwfi. In HaND, we use this field to carry the moving average of the Probe Request RSSI values that are measured by AP_{new}. On the other hand, the Capability Information fixed field currently has five reserved bits, as shown in Fig. 3.7. We use bits B9, B12, B14 and B15 (called `CHAN_BITS`) to carry the index of the channel AP_{new} operates on. The differences between regular Probe Response frames and Forwarded Response Frames and Triggering Response Frames (which will be discussed in Section 3.4.3.5) are summarized in Table 3.1.

Table 3.1 Differences between Regular, Forwarded and Triggering Probe Response Frames

Frame Type	`Length` field in DS Parameter Set	`CHAN_BITS` in Capability Information fixed field	`TRIG_BIT` in Capability Information fixed field
Regular Probe Resp. (Default)	1	all zeros	0
Forwarded Probe Resp.	RSSI measure	channel index	0
Triggering Probe Resp.	RSSI measure	channel index	1

3.4.3.4 Smart Handoff Decision with the Satisfaction-based-Fairness Heuristic

The popular *strongest-signal-first* heuristic, which ranks the candidate APs for a station (including AP_{curr} and possibly multiple neighboring APs) according to the RSSI value or its variants (such as transmission rate), has been shown to be inefficient for making handoff decisions as it may cause unbalanced loads between APs. Recent works such as [35] try to consider the AP loads in order to make proper handoff decisions. However, since none of these approaches considers the APs' detailed context information such as the transmission rates of their associated stations, they may result in performance anomaly because of the potential transmission rate diversity between the newly-associated station and the already-associated stations, as discussed in Section 3.3.4.

To deal with this issue, we propose a new heuristic for making handoff decisions, called *satisfaction-based fairness*. The key idea is to emphasize max-min fair bandwidth satisfactions among all stations. *Bandwidth satisfaction level* is not an absolute bandwidth measurement (in Mbps) but a percentage value (no unit). A mobile station may have the options of staying associated with AP_{curr} or re-associating with one of the new APs. Its handoff decision may affect its own as well as other stations' bandwidth allocations. For a certain handoff option, the bandwidth satisfaction level of a station is defined as the ratio of its expected bandwidth allocation to its maximum attainable bandwidth allocation among all possible handoff options, based on a worst-case-scenario assumption that all stations are transmitting continuously to saturate the network. In HaND, we use a simple model specified in [37], which ignores the transmission overheads such as contention window and backoff, to estimate stations' bandwidth allocations as follows. Suppose that there are n stations currently associating with an AP with the transmission rate of R_1, R_2, $\cdots$, R_n, respectively. Then, the bandwidth allocated to each station is $1/\left(\sum_{i=1}^{n} 1/R_i\right)$.

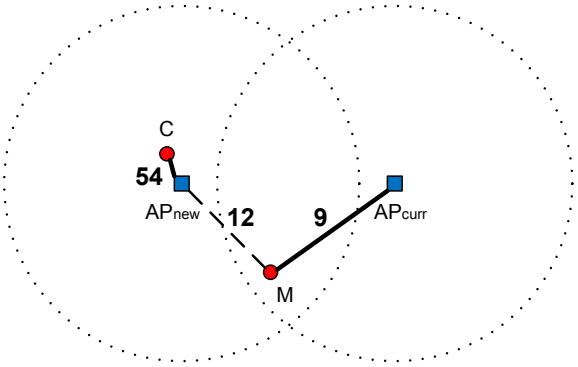
Now let's take a look at the example scenario shown in Fig. 3.8(a). AP_{curr} and AP_{new} are two APs operating on non-interfering channels, C is a static station and M is a mobile station moving from right to left. Each line in the figure represents a possible association and the number near the line represents the transmission rate (in Mbps) of the corresponding wireless link. Currently, C associates with AP_{new} and M associates with AP_{curr}. As shown in Fig. 3.8(b), when using the strongest-signal-first heuristic, M will re-associate with AP_{new}, resulting in a system throughput of 19.6 Mbps, which is only 31 percent of the maximum possible 63 Mbps. This performance anomaly occurs because the high-rate station C is "slowed down" by the newly-associated low-rate station M.

In comparison, using the proposed satisfaction-based-fairness heuristic, the maximum attainable bandwidth allocation for C and M is 54 Mbps ($= \frac{1}{(1/54)}$, which is achieved when M associates with AP_{curr}) and 9.8 Mbps ($= \frac{1}{(1/54)+(1/12)}$, which is achieved when M associates with AP_{new}), respectively. As a result, as shown in Fig. 3.8(c), M will stay associated with AP_{curr} as this will result in a higher minimum bandwidth satisfaction level among C and M, which also results in a higher overall system throughput.

To implement HaND with the proposed satisfaction-based-fairness heuristic, two pieces of information are needed: (i) the transmission rates of all candidate APs' associated stations; and (ii) the potential transmission rate of the station with each candidate AP. While the former one is exchanged periodically between APs, as discussed in Section 3.4.3.2, the latter one is not directly available. Instead, AP_{curr} only has the information about the RSSI value of the station to each candidate AP. Although the potential transmission rate depends on not only the received signal strength, but also a variety of other factors such as rate adaptation scheme and channel dynamics, results in [10, 35] show that a simple mapping between RSSI values and transmission rates can yield a satisfactory performance. In HaND, we use the mapping in Table 3.2, which is obtained from our experiments and consistent with [10].

Table 3.2 Mapping between RSSI Values and Transmission Rates

RSSI Range (dB)	0 ≀ 5	5 ≀ 8	8 ≀ 12	12 ≀ 15	15 ≀ 18	18 ≀ 22	22 ≀ 27	27 ≀ 33	33 ≀ 60
PHY Rate (Mbps)	1	2	5.5	12	18	24	36	48	54



(a) network topology

AP	B_{total}	$\{R_C, R_M\}$
AP_{new} $(*)$	19.6	$\{54, \mathbf{12}\}$
AP_{curr}	63	$\{54, \mathbf{9}\}$

(b) strongest-signal-first

AP	B_{total}	$\{S_C, S_M\}$
AP_{new}	19.6	$\{\mathbf{0.18}, 1.00\}$
AP_{curr} $(*)$	63	$\{1.00, \mathbf{0.92}\}$

(c) satisfaction-based-fairness

Figure 3.8 An example to illustrate how the proposed satisfaction-based-fairness heuristic deals with the performance anomaly that exists with the strongest-signal-first heuristic. For each heuristic, candidate APs for M to associate with and the corresponding transmission rate (denoted as R) or bandwidth satisfaction level (denoted as S) are compared. The association decisions are marked with an asterisk. B_{total} is the overall system throughput.

3.4.3.5 Triggering Probe Response Frames

Once AP_{curr} decides that the station should re-associate with a new AP, it triggers the handoff procedure by sending a special *Triggering Probe Response* frame to the station. As shown in Table 3.1, Triggering Probe Response frame has the same format as Forwarded Response Frame except that the B8 bit (called `TRIG_BIT`) of the Capability Information fixed field is set to one. Upon receiving it, the station starts the re-association process.

3.5 Experimental Study

We have implemented HaND in MadWifi [20]. In this section, we evaluate its effectiveness using experimental results.

3.5.1 Experimental Setup

All the experiments are conducted on the third floor of our department building. The (logical) network topology, the hardware configurations, and the trajectory of the mobile station are shown in Fig. 3.9. We use off-the-shelf hardware instead of sophisticated equipments to conduct experiments as this makes our experimental results comparable to what users of commodity 802.11 devices may expect in realistic scenarios.

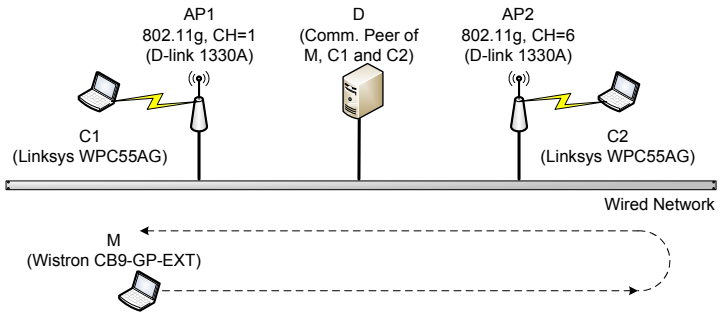
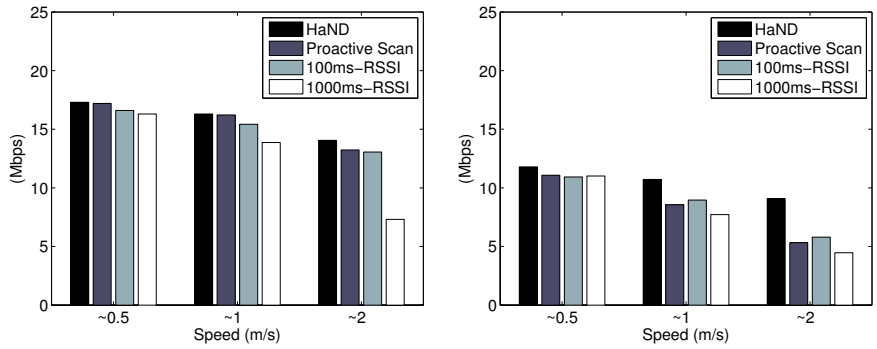


Figure 3.9 Experimental setup.

As shown in the figure, both APs operate in the 802.11g mode but on different channels (Channel 1 and Channel 6) and the wired communication backbone between them is a 100 Mbps Ethernet. Two static stations C_1 and C_2 are very close to and associate with AP_1 and AP_2, respectively. A mobile station M moves and hands off between two APs and its trajectory is shown as the dashed curve. A desktop computer D that is connected to the Ethernet serves as the communication peer for M, C_1 and C_2. We use Iperf [23] as the UDP packet generator to measure the throughput performance; CBR (Constant Bit Rate) traffic is generated with the packet size of 1470 octets. We use ICMP Echo Request packets generated by M to measure the delay performance in terms of RTT (Round Trip Time) between M and D. The SampleRate [7] rate adaptation scheme is used in all stations.

We conduct the experiments in the following scenarios. In the first set of experimental scenarios, the traffic loads on both APs are equal, while in the second set, AP_1 has heavier loads than AP_2. The results for each scenario are averaged over five experimental runs. In order to minimize potential unexpected performance variation caused by people's movement and interference from other 802.11 and Bluetooth devices, all experiments are conducted at nighttime or weekends.



(a) zero AP load (b) 20 Mbps AP load

Figure 3.10 Throughput performance of M under balanced AP loads.

3.5.2 Scenario I: Balanced AP Loads

In this scenario, two APs have the same traffic load, i.e., C_1 and C_2 generate the UDP traffic at the same application-layer rate, which varies from zero (i.e., C_1 and C_2 have no data communications with their associated APs) to 20 Mbps (i.e., both APs are heavily loaded).

We compare the performance of the proposed HaND scheme against the following schemes: (i) the *Proactive Scan* scheme proposed in [29]; and (ii) two naive handoff schemes with different scan intensity levels: *100ms-RSSI* which probes one channel per scan phase and has a scan interval of 100 ms, and *1000ms-RSSI* which probes one channel per scan phase and has a scan interval of 1000 ms. Channel dwell time is set to 5 ms in these schemes. Moreover, in these schemes, the mobile station makes the handoff decisions by ranking the candidate APs according to the RSSI value or its variants (such as transmission rate). In comparison, channel dwell time is zero in HaND and the handoff decisions are made by the mobile station's currently-associated AP.

3.5.2.1 Throughput Performance

We first compare the throughput performance of M when different testing schemes are used. M generates UDP traffic at a CBR rate of 20 Mbps and communicates with D via its associated AP. We vary the speed of M from around 0.5 m/s (slow walking) to 1 m/s (normal walking) to 2 m/s (running). Experimental results are plotted in Fig. 3.10. We have the following observations.

In general, all testing schemes yield lower throughput when (i) APs are more heavily loaded; or

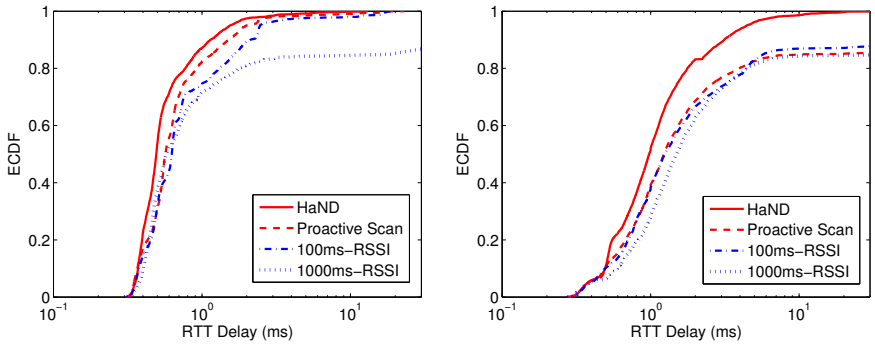
(ii) the speed of M goes up. The former one is because of the contention nature of the IEEE 802.11 protocol and M has to contend with static stations to receive the services from the AP, while the latter one is due mainly to the ineffectiveness of the SampleRate rate adaptation scheme in the presence of high fluctuation of channel conditions in mobile environments.

HaND outperforms all other schemes in all experiments, particularly when the AP load is high and/or M moves fast. As shown in Fig. 3.10(b), in the running situation under 20 Mbps AP load, the throughput of M with the proposed HaND scheme is almost twice that with other schemes. We suspect that the poor performance of Proactive Scan, 100ms-RSSI and 1000ms-RSSI may be due to the service outage experienced by M with these schemes when the AP load is high and/or M moves fast. In order to have a better understanding of this interesting phenomenon, we conduct more experiments to study the delay performances of testing schemes.

3.5.2.2 Delay Performance

We conduct the following experiment with M moving at the running speed (i.e., around 2 m/s). Instead of using the UDP data traffic, M generates ICMP Echo Request packets (at an interval of 20 ms), which provide straightforward RTT (Round Trip Time) measurements.

Fig. 3.11(a) compares the ECDFs (Empirical Cumulative Distribution Functions) of RTTs with different testing schemes when there is no loads on the APs. Both HaND and Proactive Scan perform better than others due to the dynamic adjustment of the scan interval to the link quality between M and its currently-associated AP. HaND yields better performance than Proactive Scan because of its unique zero-channel-dwell-time architecture, which reduces the service disruption period further. In comparison, 100ms-RSSI enters the scan phase at a fixed interval of every 100 ms and hence disrupts the data services more often. 1000ms-RSSI performs the worst among all testing schemes. This is because it enters the scan phase every 1000 ms. Therefore, when M moves fast, it may be slow in collecting the information about nearby APs. As a result, M may have already lost the connection to its current AP before re-associating with a new one, i.e., service outage may occur. As shown in the figure, about 14.9 percent of the packets experience infinite RTT, meaning that they are transmitted during the service outage. This explains the low throughput performance of 1000ms-RSSI in the running



(a) zero AP load (b) 20 Mbps AP load

Figure 3.11 Delay performance of M under balanced AP loads.

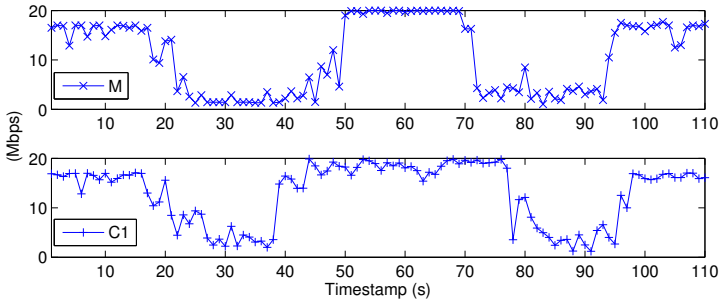
situations as shown in shown in Fig. 3.10(b).

In Fig. 3.11(b), the AP load is increased to 20 Mbps and the service outage phenomenon can be observed more clearly from the experiments. When the AP is heavily loaded, in addition to 1000ms-RSSI, Proactive Scan and 100ms-RSSI also experience service outage as well. The reason is as follows. Service outage may have two different causes: (i) it may be caused by a long channel scan interval, as explained in the previous paragraph for 1000ms-RSSI; or (ii) it also may occur if the Probe Responses are not received by the station. In our experiment, a short channel dwell time of 5 ms may not be long enough for M to collect the returning Probe Responses from the AP since the AP is busy serving other stations with a load of 20 Mbps. In comparison, HaND continues to perform well (with zero service outage) as it does not rely on dwelling on a new channel to collect Probe Response frames; instead, it leverages on the wired communication backbone between APs to relay the information about the new AP through the station's currently-associated AP. This is one of the key advantages of the proposed HaND scheme.

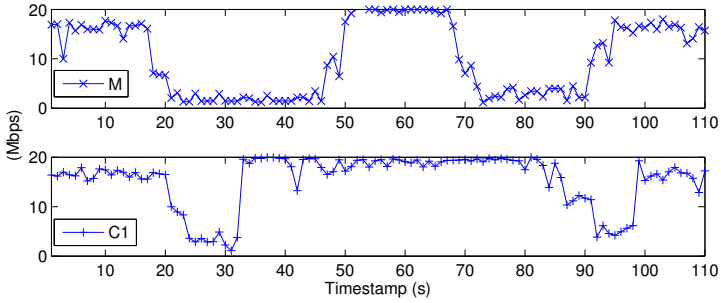
3.5.3 Scenario II: Unbalanced AP Loads

In this section, we study the performance of the proposed satisfaction-based-fairness heuristic used by a mobile station's current AP to make the handoff decisions. In the experiment, two APs are set up with unbalanced loads as follows: C_1 generates the UDP traffic at a CBR rate of 20 Mbps while C_2 remains idle all the time. We vary the speed of M from around 0.5 m/s to 2 m/s. We compare the

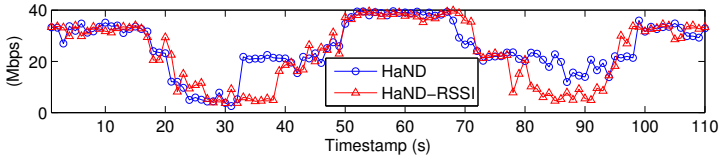
performance of HaND with a variant of HaND called HaND-RSSI, which differs from HaND in the handoff triggering heuristic. In HaND-RSSI, the station's currently-associated AP makes the handoff decisions based on the RSSI values of the candidate APs.



(a) Snapshot of instant throughout of HaND-RSSI



(b) Snapshot of instant throughout of HaND



(c) System instant throughout (Mbps)

Figure 3.12 Snapshots of instant throughput under unbalanced AP loads. The speed of the mobile station M is around 0.5 m/s.

Fig. 3.12 plots the snapshots of the instant throughputs of M and C_1 as well as the overall system instant throughput for HaND and HaND-RSSI, respectively, when M moves at around 0.5 m/s. Results with HaND-RSSI are shown in Fig. 3.12(a). Initially, M is close to and associates with AP_1. As it moves away from AP_1, its transmission rate decreases as dictated by the adopted SampleRate rate adaptation scheme. As a result, the instant throughput of M starts dropping. Meanwhile, the instant throughput of C_1 also starts dropping since the the IEEE 802.11 protocol is designed to provide equal access to the shared channel among stations associating with the same AP. At around the 40-second mark, the signal strength of AP_2 becomes higher than that of AP_1. Hence, M triggers a handoff and

Table 3.3 Comparison of System Throughput (in Mbps) under Unbalanced AP Loads

Speed of M	∼0.5 m/s	∼1 m/s	∼2 m/s
HaND	27.49	25.86	18.66
HaND-RSSI	23.48	22.51	18.31

re-associates with AP_2. Now M and C_1 associates with different APs without contending with each other. As a result, the instant throughput of both stations increases immediately. This can be seen clearly from the figure. M continues to move and at around the 78-second mark, M triggers another handoff and re-associates with AP_1, resulting in another drop of the instant throughput of C_1. As M moves closer to AP_1, the link quality between them gets better. At around the 98-second mark, both M and C_1 transmit at high rates and receive high-throughput data services.

In comparison, the instant throughputs of HaND are shown in Fig. 3.12(b). Since M's currently-associated AP makes the handoff decisions based on the bandwidth satisfaction level, we observe that when M moves away from AP_1, a handoff is triggered at an earlier time instance than HaND-RSSI at approximately the 32-second mark. This is because at that moment, AP_1 realizes that M has become a low-rate station and it is a better option to handoff M to AP_2 to prevent M from "slowing down" the high-rate station C_1. As a result, between 32 and 40 seconds, HaND yields a higher system throughput than HaND-RSSI, which is shown in Fig. 3.12(c). Similarly, when M moves back towards AP_1, HaND triggers a handoff later than HaND-RSSI, resulting a higher system throughput between 78 and 86 seconds as shown in the figure.

We also compare the system throughput performance of HaND and HaND-RSSI at different speeds of M in Table 3.3. We observe that with slow or normal mobility (at 0.5 or 1 m/s), HaND outperforms HaND-RSSI by about 17%. However, it is interesting to see that, with high mobility (at 2 m/s), the performances of HaND and HaND-RSSI are similar. This is probably due to the following reasons: (i) the time period during which HaND outperforms HaND-RSSI shrinks as the moving speed increases; and (ii) the default SampleRate rate adaptation scheme is unable to track the dynamic channel variation well in mobile environments. The latter issue has been recognized by several works on rate adaptation. We expect that a larger performance gain may be achieved if the station adopts a more responsive rate adaptation scheme to work with the proposed HaND scheme.

3.6 Conclusions

From experiments, we find that channel dwell time contributes to a major part of the handoff delay and it is difficult to set a proper channel dwell time for an 802.11 station since the context information about the scanned channel, such as the number of APs on the channel and the loads of the APs, is unavailable to the station. To address this challenge, we propose a practical fast handoff scheme, called HaND, which adopts a novel zero-channel-dwell-time architecture to reduce the handoff delay. Moreover, in HaND, the handoff decisions are made by a station's currently-associated AP (rather than the station itself) based on a new satisfaction-based-fairness heuristic. We have implemented HaND in MadWifi and experimental results show that HaND outperforms other testing schemes in terms of both throughput and delay.

CHAPTER 4. SMART AND SEAMLESS LOAD BALANCING WITH MULTIPLE INTERFACES

4.1 Introduction

Currently, Wi-Fi hotspots are one of the leading ways for individuals to connect to the Internet. Clients may use a variety of devices, including smart phones, laptops, tablets, etc., to connect to the Internet via Wi-Fi hotspots from a variety of commercial locations, including airports, hotels, retail locations, and coffee shops. However, as the popularity of Wi-Fi increases, the demand for bandwidth at each access point also increases. For instance, a Wi-Fi network in an airport may have hundreds of users such that a single AP would be insufficient to satisfy the network demand. A naive solution may deploy multiple APs throughout the airport terminal such that each AP serves a fewer number of users. However, even with enhancements to coordinate associations among APs, this solution may be insufficient in the long term for a number of reasons. Firstly, such a system must determine which AP each user connects to in an effort to maximize system performance. Secondly, because some form a dynamic association scheduling would be necessary, user reassociation might be frequent. This is a time-consuming process, as a station must disassociate from its current AP and then authenticate and reassociate with a new AP. Thirdly, the multiple APs must exchange information regarding the connected users to aid in association scheduling, which adds a communication overhead to the system. Finally, such a system may have issues routing downlink traffic to stations which perform a reassociation, causing packets to be delayed or lost.

To mitigate the above mentioned issues, we propose a unique solution called SAP (Smart Access Point) which utilizes a One-AP-Multiple-Interface (OAMI) architecture to balance network load across multiple interfaces and to dynamically provide association scheduling for connected clients. Specifically, SAP has the following features:

- SAP introduces an *Interface Unification Layer* to unify the appearances of multiple interfaces, which makes seamless handoff a possibility.
- SAP utilizes a *Shared Node Table* to facilitate fast reassociation between interfaces. Reassociation can be performed simply by copying the station context to another interface and alerting the client of a BSS channel switch.
- SAP mitigates issues with downlink traffic via a *Bridge Forwarding Module* which allows packets to be forwarded to the correct interface and their final destination.
- SAP has a *Monitoring & Execution Module* which monitors various statistics for each connected station (including the traffic rate) and reports them to a *Coordination Module* to determine association scheduling.
- SAP is an AP-only solution without any modifications at the stations or external routers.

The rest of the paper is organized as follows. Observations motivating SAP are described in Section 4.3, along with accompanying design goals. The design of the SAP architecture is detailed in Section 4.4. Section 4.5 introduces a network model to calculate the maximum service rate of a station. We present experimental based performance evaluation results in Section 4.6. Related work is discussed in Section 4.2 and we conclude the paper in Section 4.7.

4.2 Literature Survey

Related work in the literature regarding association control in IEEE 802.11 networks can be classified into two main categories, namely *handoff delay optimization* and *association metric design*.

4.2.1 Handoff delay optimization

Various fast handoff schemes have been proposed in the past with the purpose of reducing various components of handoff delay. In [25], the authors showed that the scan phase is the most significant contributor to the overall handoff delay and the variations in channel dwell time account for the large variations in the handoff delay. The schemes proposed in [26, 27] try to reduce the number of channels

to probe during the scan phase. For example, [27] adopts a neighbor graph approach and the station only probes the channels that have active APs operating on. D-Scan [28], Proactive-Scan [29], [27], and [30] try to reduce channel dwell time. SyncScan [31] reduces the probe duration by having all APs synchronize their Beacon frames in the case of passive scan. D-Scan, Proactive-Scan and the scheme in [32] interleave the long scan phase to reduce the packet delay. Different from the above schemes which all try to reduce the scan delay, the scheme in [33] aims to reduce the reassociation delay. HaND [38] removes the channel dwell time in scan delay via letting AP make the handoff decisions for stations. Compared with the previous works, SAP eliminates the scan delay, the authentication delay and the reassociation delay entirely, thus reducing the handoff delay to the minimum.

4.2.2 Association metric design

Another research topic in this area is how to decide the station-to-AP association pattern. One of the commonly-used metrics is the signal strength. For example, hysteresis-based approaches in [28,29,32] direct a station to reassociate with the new AP that has a stronger signal strength. [26] compares several association metrics that are based on signal strength statistics. [34] combines the signal strength value with the large-time-scale performance measurement. In [35], another metric is proposed which is called available capacity. [37–41] consider the fairness among all stations in the association control. Specifically, [37–39] propose a throughput fulfillment factor. [41] provides air time fairness among stations. By considering the traffic rate instead of saturation analysis, [40] adopts a proportional fairness metric taking the station's traffic need into account, and evaluate it via simulation only. In comparison, SAP proposes a new association metric by considering stations' current traffic rates, and evaluates it with real-world experiments.

4.3 Motivations and Design Goals

In this section, we discuss the emerging One-AP-Multiple-Interface architecture for Wi-Fi networks and present a few interesting observations from experiments regarding the station behaviors in Wi-Fi networks, which motivate us to design and implement the proposed SAP architecture. Experiments are conducted with Dell Optiplex desktop and Latitude laptops equipped with wireless adapters which

embed Atheros 5212 chipset. Each machine is loaded with a MadWifi device driver [20] to collect the experimental data.

4.3.1 Motivations and Observations

4.3.1.1 One-AP-Multiple-Interface Architecture

As Wi-Fi becomes more ubiquitous, in order to serve more Wi-Fi stations in a network, it is a more cost-effective solution to equip an AP with multiple interfaces operating on different channels than to deploy multiple APs in the network. However, by simply adding more interfaces to an AP without carefully planning how the interfaces shall work together to serve the stations, the OAMI (One-AP-Multiple-Interface) architecture is merely "multiple APs in the same box," which is shown in Fig. 4.3(a).

Innovations are needed to coordinate between multiple interfaces so stations can truly enjoy the benefits that OAMI offers over the conventional OAOI (One-AP-One-Interface) architecture. For example, as multiple interfaces reside on the same machine under OAMI, the handoff delay of moving a station from one interface to another under OAOI (which will be discussed next) could be minimized. In another example, the coordination between interfaces under OAMI could be accomplished internally without introducing extra communication overhead to the backbone network (such as Ethernet) that connects the interfaces (on different APs) under OAOI.

4.3.1.2 Long Handoff Delay under OAOI

The IEEE 802.11 handoff process (to move a station from an AP to another) could be very time-consuming [25, 38]. During the process, the station needs to perform channel scanning, authentication and reassociation. Moreover, additional procedures that may be needed at higher layers such as DHCP discovery could render the entire process even longer.

We examine how large the handoff delay could be with a simple experiment that uses two APs and one station. The station keeps sending Ping packets to a remote server, once every 200 ms. At the 8-second mark, the station starts the handoff process to associate with the other AP. Fig. 4.1 plots the RTT (Round Trip Time) of the Ping packets. As shown in the figure, the entire handoff process takes

about six seconds to complete, during which the station experiences severe service disruption; this is evidenced by the drastically increased RTTs of Ping packets during the process.

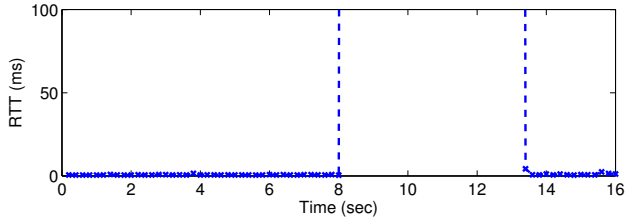


Figure 4.1 Large RTT of the Ping packets during the legacy handoff process.

In practice, it is always desirable to have a seamless handoff process with minimal handoff delay and thus eliminate the service disruption, which could be critical to many user applications such as real-time video streaming.

4.3.1.3 Revisit of Performance Anomaly

Performance anomaly is a well-known phenomenon in 802.11 networks [36]. It is caused by the transmission rate diversity between all stations associated with the same AP. With rate diversity, the high-rate station is "slowed down" by the low-rate station because the 802.11 protocol is designed to allow all contending stations to access the channel with an equal probability. Due to throughput loss of performance anomaly, many association scheduling schemes try to avoid it by considering the transmission rates when making association decisions. For example, stations may be grouped according to their transmission rates; all the high-rate stations are associated with one AP while another AP serves all the low-rate stations.

One of the key assumptions in the performance anomaly study is that all contending stations are saturated or heavily loaded. We now revisit the performance anomaly in practical scenarios where the low-rate station may be under various loads. In the experiment, we associate two stations with the same AP; one of them communicates with the AP at the low 1 Mbps rate while the other at the high 54 Mbps rate. The high-rate station is always saturated while the load of the low-rate station increases gradually from 10 Kbps to 2000 Kbps.

As shown in Fig. 4.2, when the low-rate station is lightly loaded, the throughput of the high-rate

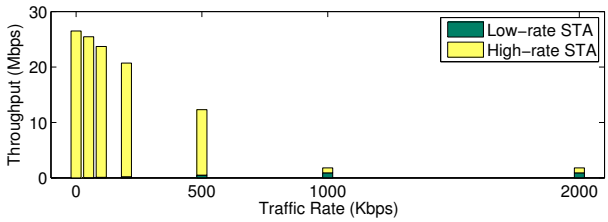


Figure 4.2 Throughput comparison when the low-rate (1 Mbps) station is under different loads. Two stations are used in the experiment and the high--rate (54 Mbps) station is always saturated.

station is less affected as the low-rate station only contends for the channel sporadically. As the load of the low-rate station increases, the performance anomaly becomes more apparent. When the low-rate station becomes saturated with a load of 1000 or more packets per second, the classic performance anomaly appears with two stations receiving an equal throughput and the overall system throughput suffering a significant drop.

In practice, as stations could be very diverse in terms of their traffic loads, it is important to consider the actual load condition of each station (whenever possible) when making the association decisions.

4.3.2 Design Goals

Based on the above observations, we propose in this work a SAP (Smart Access Point) realization of the OAMI architecture. With SAP, the AP shall be able to:

- adjust the association scheduling decisions between multiple interfaces dynamically based on stations' actual and varying traffic load conditions; and
- execute the association scheduling decisions with seamless handoff of stations between interfaces with minimal handoff delay, by taking full advantage of the OAMI architecture.

Moreover, SAP shall be an AP-only solution. It shall not require any modification at the client side and shall work with all legacy stations, thus facilitating its practical deployment.

4.4 The Proposed SAP Architecture

In this section, we first present an overview of the proposed SAP architecture, then describe each SAP module in detail.

4.4.1 SAP Overview

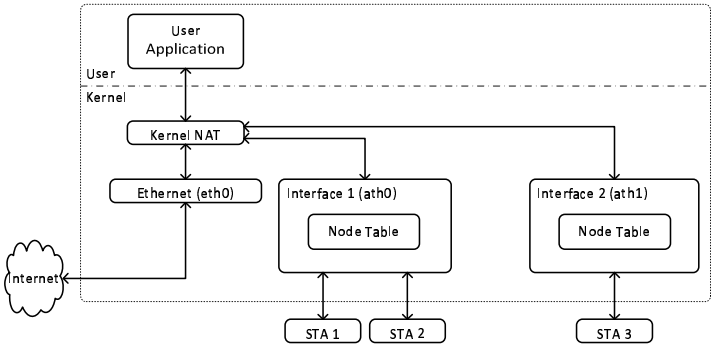
The overall structure of the proposed SAP architecture is shown in Fig. 4.3(b). In comparison to the basic OAMI architecture in Fig. 4.3(a), SAP has the following additional function blocks: (i) *Interface Unification Layer*; (ii) *Shared Node Table*; (iii) *Bridge Forwarding Module*; (iv) *Monitoring & Execution Module*; and (v) *Coordination Module*. All these modules are implemented at the AP only, without the need of any modifications at the stations or external routers.

(i) *Interface Unification Layer*. SAP creates an Interface Unification Layer to unify the appearances of multiple interfaces, which makes the handoff process transparent to the stations and thus seamless handoff a possibility. Details will be discussed in Section 4.4.2.1.

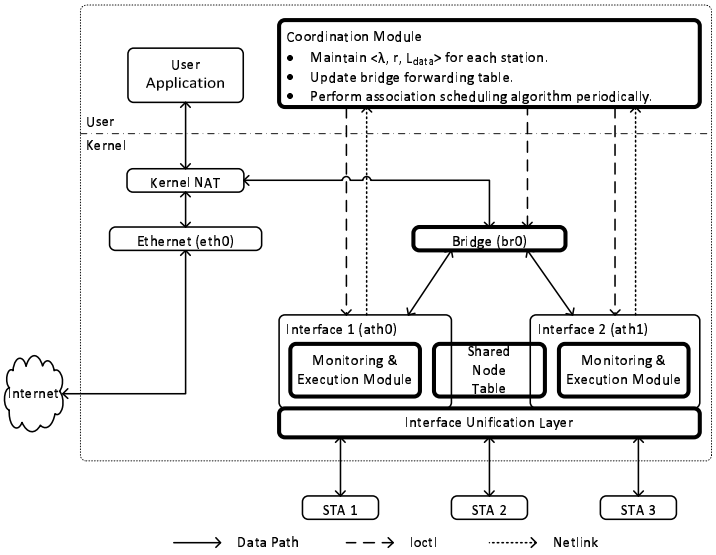
(ii) *Shared Node Table*. Different from OAMI, where each interface maintains a separate table of the stations that are associated with it, SAP allows multiple interfaces to share a single node table. This simplifies significantly the context transfer procedure that is required by the handoff process. Details of the Shared Node Table structure will be discussed in Section 4.4.2.2.

(iii) *Bridge Forwarding Module*. In OAMI, after handoff, to make sure proper forwarding of a station's downlink traffic through the new interface to reach the station, the routing table at an external router needs to be updated. In comparison, SAP sets up an internal bridge for this purpose. Details of the Bridge Forwarding Module will be discussed in Section 4.4.2.3.

(iv) *Monitoring & Execution Module*. In SAP, each interface monitors the statistics of all the stations that are associated with it, including the physical transmission rates, the traffic rates, and the data payload sizes. Collected statistics are reported to the *Coordination Module* that uses them as inputs to its association scheduling algorithm. Meanwhile, the Monitoring & Execution Module also accepts commands from the Coordination Module and executes them by sending the corre-



(a) The basic OAMI architecture.



(b) Our proposed SAP architecture.

Figure 4.3 Comparison of our proposed SAP architecture and the basic OAMI architecture. Function blocks added in SAP are marked in bold.

sponding management frames to the station if needed. Details of this module will be discussed in Section 4.4.2.4.

(v) *Coordination Module*. The Coordination Module in the user space coordinates the behaviors of all modules discussed above. It accepts the statistics reported by the Monitoring & Execution Modules of multiple interfaces, and runs an association scheduling algorithm periodically to determine which station should be moved to another interface. Based on the output of the algorithm, it issues commands to the Monitoring & Execution Module, and updates the Bridge Forwarding Table accordingly. Details will be discussed in Section 4.4.2.5.

4.4.2 SAP Modules

This section describes the five modules of SAP in detail.

4.4.2.1 Interface Unification Layer

In the basic OAMI architecture, different interfaces (i) have different MAC addresses, (ii) operate on different non-overlapping channels, and (iii) may have different BSSIDs. Therefore, in order for a station to continue receiving services from the AP via a different interface than the current one it is associated with, it needs to not only switch to the channel that the new interface operates on, but also use the BSSID and MAC address of the new interface when preparing its packets; otherwise, all its packets will simply be discarded by the new interface due to mismatch of information. For this reason, the conventional handoff process typically consists of the following steps [1]:

(i) The station performs channel scanning;

(ii) The station extracts the BSSID and MAC address of the new interface from Beacon or Probe Response frames received;

(iii) The station switches to the channel that the new interface operates on;

(iv) The station exchanges management frames with the new interface to complete authentication and reassociation processes;

(v) The station prepares its packets using the BSSID and MAC address of the new interface and starts communicating with the AP via the new interface.

As we can see, the station is required to participate actively in the entire handoff process, where various delays occur at various steps of the process.

To simplify the handoff process and reduce the handoff delay, SAP introduces an Interface Unification Layer (IUL). It unifies all interfaces and makes them appear the same (except that they operate on different non-overlapping channels) to the stations by setting the same BSSID and the same MAC address for all interfaces. This could be done via, for example, the `ioctl` command (to set the BSSID) and the `ath_attach()` function (to set the MAC address) in MadWifi. As a result, Step (v) of the

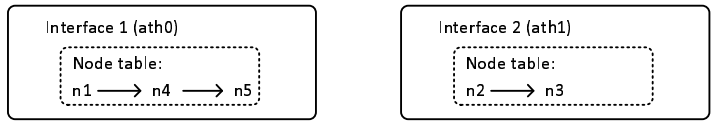
handoff process could be avoided completely. As we will discuss next, with the help from the Shared Node Table structure and the Bridge Forwarding Module, which help remove Step (iv), and the Monitoring & Execution Module together, which help remove Steps (i) and (ii), the handoff process is simplified to a channel switch operation, thus reducing the handoff delay to the minimum.

4.4.2.2 Shared Node Table

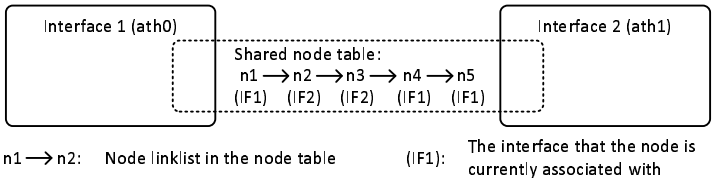
As we mentioned in Section 4.4.2.1, in the basic OAMI architecture, a station has to perform authentication and reassociation before being served again by the new interface. The root reason for this requirement is that, under OAMI, each interface maintains a separate node table that keeps track of only the stations that are associated with it, as shown in Fig. 4.4(a). Therefore, the context transfer procedure during the handoff process under OAMI consists of the following steps:

- Firstly, the station disassociates from the current interface which then removes the corresponding entry from its node table;
- Then, the station scans, authenticates and reassociates with the new interface;
- Finally, the new interface creates a new entry in its node table, which is populated by the context information of the station, such as security information, association information, and so on.

As we can see, the station has to go through the time-consuming authentication and reassociation in order to create an entry in the new interface's node table.



(a) Default node table structure.



(b) Shared node table structure in SAP

Figure 4.4 Comparison of the shared node table structure in SAP and the default node table structure.

The context transfer procedure described above does not take advantage of the fact that the interfaces reside on the same machine under OAMI, which means that they are served by the same kernel and the same memory space. Hence, it is possible to simplify the context transfer procedure via internal actions, which is exactly what SAP does. Specifically, SAP maintains a single node table shared by all interfaces, as shown in Fig. 4.4(b). Each entry in the shared node table now has an extra flag to indicate the interface that the station is currently associated with. With such a shared node table structure, the context transfer procedure is simplified to a simple modification of a flag in the shared node table. As a result, authentication and reassociation are avoided completely.

4.4.2.3 Bridge Forwarding Module

In the basic OAMI architecture, upon a successful handoff of a station to a new interface, the external router needs to update its routing table so the downlink packets for the station could be routed properly toward the new interface. SAP uses a different forwarding mechanism to achieve this purpose. It is an internal approach that limits all the modifications within the AP only.

Specifically, SAP establishes an internal bridge that interconnects the multiple interfaces at the data link layer, and each interface is assigned a bridge *port* number. The bridge has a *forwarding table* which contains entries for each station that is associated with the AP. Each entry contains the following information: (i) the MAC address of the station; and (ii) the bridge port number of the interface that the station is associated with. As shown in Fig. 4.3(b), all the downlink packets are first routed to the bridge, and then forwarded to the appropriate interface by looking up the bridge forwarding table. Upon a successful handoff of a station, SAP updates the corresponding entry in the bridge forwarding table, in addition to the corresponding entry in the Shared Node Table discussed in Section 4.4.2.2.

4.4.2.4 Monitoring & Execution Module

In SAP, each interface monitors the stations that are associated with it and collects the station statistics for facilitating the association scheduling algorithm in the Coordination Module. The collected statistics include the downlink and uplink traffic rates, the transmission rate and the data payload length of each station.

Based on the collected statistics, each interface reports the following information to the Coordination Module periodically: λ, r and L_{data}. Here, λ is the overall traffic rate of the station, and we use the uplink throughput to approximate the uplink traffic rate of the station. r is the communication rate between the station and the AP. L_{data} is the data payload length of the station. In our implementation of SAP in the MadWifi device driver, we create `netlink` sockets to enable the information exchange between the Monitoring & Execution Module (in the system kernel) and the Coordination Module (in the user space).

Another function of the Monitoring & Execution Module is that it accepts an `ioctl` command from the Coordination Module and executes it by sending a corresponding management frame to the station. For example, if the Coordination Module determines that the station with a MAC address of `00:1c:f0:f9:a3:4c` should be moved from Interface `ath0` that the station is currently associated with to a different interface that operates on Channel 11, it issues the following `ioctl` command:

```
iwpriv ath0 switch 00:1c:f0:f9:a3:4c 11
```

to Interface `ath0`. Subsequently, the Monitoring & Execution Module of Interface `ath0` sends an `Channel Switch Announcement (CSA)` management frame to notify the station to switch to Channel 11. Note that `Channel Switch Announcement (CSA)` is a standard management frame defined in the 802.11h [42] standard and its original purpose was to allow an AP to notify all its associated stations of the BSS channel change after it changes its own operating channel. Once the station receives the management frame, it will follow the standard to switch to the new channel. The creative use of the `Channel Switch Announcement (CSA)` management frame is another key factor that contributes to SAP's seamless handoff process.

Seamless Handoff in SAP So far, we have discussed the Interface Unification Layer, the Shared Node Table structure, the Bridge Forwarding Module and the Monitoring & Execution Module, which are the four key SAP components that work together to enable seamless handoff. To summarize, with SAP, the handoff of a station between interfaces is simplified to

- two simple internal AP actions to update the Shared Node Table (i.e., modification of a flag) and

the Bridge Forwarding Table (i.e., modification of an entry); and

- a simple `ioctl` command from the AP to the old interface which then sends a management frame to notify the station of the channel switch.

Comparing with the conventional handoff process that consists of channel scanning, channel switch, station authentication, station reassociation and the accompanying upper layer actions such as DHCP discovery, the handoff delay in SAP has been reduced to the minimum: the channel switch time.

4.4.2.5 Coordination Module

SAP's Coordination Module resides in the user space. It collects the statistics reported by the Monitoring & Execution Modules of multiple interfaces, and uses them as inputs to an association scheduling algorithm. The goal of the algorithm is to find the *association pattern* – which specifies the station-to-interface association for all stations – that provides the best user experience to all stations.

In the past, many performance metrics have been proposed to quantify the user experience, such as throughput, airtime, and so on. In this paper, we define a performance metric called *traffic fulfillment* (TF) and use the minimum traffic fulfillment among all stations as a quantitative measure of the user experience. Note that SAP is not limited or bound to this particular metric and can work fine with other performance metrics as well by adjusting the association scheduling algorithm accordingly. For a given association pattern, *traffic fulfillment* (TF) of a station is defined as:

$$\text{TF} = \frac{G}{\min\{r, \lambda\}}, \tag{4.1}$$

where G is the actual throughput obtained by the station, r is the transmission rate between the station and the AP, and λ is the traffic rate required by the station. TF is defined this way to reflect how well the station's requested traffic rate can be fulfilled by the system, which could be a good measurement of the user experience. Clearly, different association patterns may result in different G. We will discuss how to estimate G in Section 4.5.

Formally, the optimization problem that the association scheduling algorithm tries to solve is defined as follows. Given the transmission rates (r), the traffic rates (λ), and the data payload lengths

(L_{data}) of all stations, the goal is to find the best association pattern that maxi-minimizes the traffic fulfillment (TF) among all stations:

$$\text{pattern}^{\text{OPT}} = \arg \max_{\{\text{all association patterns}\}} \text{TF}_{\min}, \tag{4.2}$$

where $\text{TF}_{\min} = \min_i \text{TF}_i$ is the minimum TF among all n stations given an association pattern. Note that, under certain circumstances, there might exist multiple association patterns that all produce a high level of traffic fulfillment to all stations, e.g., $\text{TF}_{\min} \geqslant 0.9$. We call these association patterns *candidate patterns*. A secondary goal of the algorithm is to choose from the candidate patterns the one that maxi-minimizes the *maximum service rate* (MSR) among all stations:

$$\text{pattern}^{\text{OPT}} = \arg \max_{\{\text{all candidate patterns}\}} \text{MSR}_{\min}, \tag{4.3}$$

where $\text{MSR}_{\min} = \min_i \text{MSR}_i$. Here, the *maximum service rate* (MSR) is defined as the maximum throughput that a station may obtain if it increases its traffic rate to infinity, given that the status remains unchanged for all other stations competing with this station for the same interface. The secondary goal is defined this way to ensure that SAP selects the candidate pattern that is most capable of handling a sudden increase of traffic rate requirement from an arbitrary station. We will discuss how to calculate MSR in Section 4.5.

The optimization problems in Eqs. (4.2) and (4.3) have been proved to be NP-hard [37]. Many approximation algorithms have been proposed in the literature to solve similar problems. As the main focus of this paper is not to design a better approximation algorithm, we propose a simple and practical greedy solution to be used in the association scheduling algorithm; evaluation results in Section 4.6 show that great performance gain can be achieved even with such a simple solution. The idea is that, instead of searching over all possible association patterns (which grows exponentially with n – the total number of stations in the network), the association scheduling algorithm switches at most one station to a different interface during each scheduling period. This way, the search space is effectively reduced to $n(m-1)$ where m is the number of interfaces, hence improving the scalability of the algorithm significantly. Algorithm 1 gives the pseudo-code of the proposed solution for an AP with

two interfaces. As shown in the pseudo-code, during each scheduling period, the algorithm determines whether or which station should be switched to the other interface so that $\mathrm{TF}_{\min}$ can be increased the most. If the algorithm finds one successfully, the Coordination Module issues an `ioctl` command to the corresponding interface, which in turn notifies the station to make the channel switch, as already discussed in Section 4.4.2.4.

Algorithm 1 The Association Scheduling Algorithm in SAP

INPUT: pattern$^{\text{curr}}$: $\vec{\mathbf{t_1}}$ associated with IF_1, $\vec{\mathbf{t_2}}$ associated with IF_2
OUTPUT: station x^* that should be switched to the other interface
/* Initialization*/
$\mathrm{TF}_{\text{maximin}} = \mathrm{TF}_{\min}$ of the current association pattern
$\mathrm{MSR}_{\text{maximin}} = \mathrm{MSR}_{\min}$ of the current association pattern
pattern$^{\text{TF}}$ = pattern$^{\text{curr}}$; pattern$^{\text{MSR}}$ = pattern$^{\text{curr}}$
for each x in $\vec{\mathbf{t_1}} \cup \vec{\mathbf{t_2}}$ **do**
    Consider the new association pattern by switching x
    Calculate $\mathrm{TF}_{\min}$ and $\mathrm{MSR}_{\min}$
    if $\mathrm{TF}_{\min} > 1.1 * \mathrm{TF}_{\text{maximin}}$ **then**
        $\mathrm{TF}_{\text{maximin}} = \mathrm{TF}_{\min}$; record pattern to pattern$^{\text{TF}}$
    end if
    if $\mathrm{TF}_{\min} \geqslant 0.9$ **then**
        if $\mathrm{MSR}_{\min} > 1.1 * \mathrm{MSR}_{\text{maximin}}$ **then**
            $\mathrm{MSR}_{\text{maximin}} = \mathrm{MSR}_{\min}$
            record pattern to pattern$^{\text{MSR}}$
        end if
    end if
end for
if $\mathrm{TF}_{\text{maximin}} \geqslant 0.9$ **then**
    x^* = set-diff(pattern$^{\text{MSR}}$, pattern$^{\text{curr}}$)
else
    x^* = set-diff(pattern$^{\text{TF}}$, pattern$^{\text{curr}}$)
end if

4.5 Modeling and Analysis

In this section, we present a simple model to calculate the maximum service rate (MSR) of a station.

4.5.1 Calculation of MSR with Two Contending Stations

We first consider the basic scenario when two and only two stations (s and t) are associated with the same interface and contend for channel access. The transmission rate, the traffic rate and the data payload length of a station are denoted as r, λ and L_{data}, respectively. By definition, the MSR of station s is the maximum throughput that s may obtain when it increases λ_s to ∞ (i.e., becomes saturated), given that the traffic rate of station t (λ_t) remains unchanged.

Similar to [40, 43, 44], we use *virtual slot* to characterize the airtime usage, which is defined as a full transmission cycle including backoff, data transmission time and ACK transmission time, or a single slot time if no station is contending for the channel. Let q denote the probability that a station contends for the channel at the end of a virtual slot. Clearly, when calculating the MSR of station s, q_s is always one as s is assumed to be saturated. Therefore, we have approximately the following:

$$\begin{aligned} P_s &= q_s(1-q_t) + \frac{1}{2}q_s q_t = 1 - \frac{1}{2}q_t, \\ P_t &= q_t(1-q_s) + \frac{1}{2}q_s q_t = \frac{1}{2}q_t, \end{aligned} \tag{4.4}$$

where P_s and P_t are the probability that s and t completes a transmission successfully in a virtual slot. Eq. (4.4) is based on the assumption that two stations have an equal probability of 0.5 to seize the channel successfully when they both contend in the same virtual slot. We derive q_t based on a Markov chain model we have proposed to characterize the behaviors of station t. It has $N+1$ states where N is the maximum queue length allowed at t. State i ($0 \leqslant i \leqslant N$) of the Markov chain represents that t's queue length is $Q_t = i$ at the end of a virtual slot. Thus, by definition, we have $q_t = P(Q_t \neq 0) = 1 - \pi_0$ where π_0 is the steady state probability of state 0. Plugging q_t back to Eq. (4.4), P_s and P_t can then be derived.

As shown in Fig. 4.5, we define a Markov chain with $N+1$ states for station t, where N is the maximum queue length allowed at t. State i ($0 \leqslant i \leqslant N$) of the Markov chain represents that t's queue length is $Q_t = i$ at the end of a virtual slot. In this Markov chain, state i can only transit to state j where $j \geqslant i-1$, as the station can transmit at most one packet in a virtual slot. In particular, state i transits to state $i-1$ if t transmits successfully in the virtual slot and there are no new packet arrivals

at t during the transmission. The state transition probabilities are:

$$P_{j|i} = P\{Q_t = j | Q_t = i\} = \begin{cases} \delta(T_s, j), & 0 = i \leqslant j < N, \\ 0, & 1 \leqslant j+1 < i \leqslant N, \\ \frac{1}{2}\delta(T_t, 0), & 1 \leqslant j+1 = i \leqslant N, \\ \frac{1}{2}\delta(T_s, j-i) & \\ +\frac{1}{2}\delta(T_t, j-i+1), & 1 \leqslant i \leqslant j < N, \\ \frac{1}{2}\sum_{k=N-i}^{\infty}\delta(T_s, k) & \\ +\frac{1}{2}\sum_{k=N-i+1}^{\infty}\delta(T_t, k), & 0 \leqslant i \leqslant j = N, \end{cases} \tag{4.5}$$

where T is the effective transmission duration of a station, including backoff, data transmission time, and ACK transmission time. $\delta(T, n)$ is the probability that n packets arrive at station t during the time interval T. For example, in the case of Poisson arrivals, $\delta(T, n) = \frac{(\lambda_t T)^n}{n!} e^{-\lambda_t T}$.

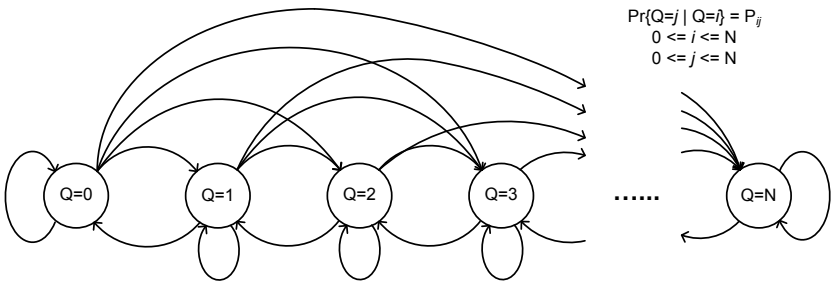


Figure 4.5 The Markov chain model for station t, which is competing with a saturated station s.

The meanings of the transition probabilities are explained next. The first equation in Eq. (4.5) states that at state 0, t has no packet in its queue. Therefore, it will transit to state j if there are j packet arrivals during station s's effective transmission time T_s since s is saturated and always ready to transmit. The second equation states that state i can never transit to state j if $j < i - 1$. This is because the station can transmit at most one packet in a virtual slot. The third equation means that, in order to transit from state i to state $i - 1$, the following conditions need to be satisfied: (i) t competes with s and seizes the channel successfully in the virtual slot, with a probability of 0.5; and (ii) no new packets arrive during T_t. The fourth equation states that a transition from state i to j ($j \geqslant i$) occurs when either (i) s seizes

the channel and there are $j - i$ packet arrivals during T_s; or (ii) t seizes the channel and transmits one packet, and there are $j - i + 1$ packet arrivals during T_t. The fifth equation describes a special case caused by the buffer size of N. That is, if $Q_t > N$ at the end of a virtual slot, t simply drops the extra packets and the Markov chain stays at state N.

With the transition probabilities of the Markov chain defined, we can calculate the steady state probability of state 0, i.e., π_0. Then, by definition, we have $q_t = P(Q_t \neq 0) = 1 - \pi_0$; plugging it back to Eq. (4.4), P_s and P_t can then be derived. With P_s and P_t, MSR of station s can be calculated with Algorithm 2.

Algorithm 2 Calculation of MSR of s given that only a single station t contends with s.

1: **INPUT:** s, t and their status
2: **OUTPUT:** MSR(s, t)
3: Calculate T_s and T_t based on r_s, $L_{\text{data}}(s)$, r_t, $L_{\text{data}}(s)$
4: Calculate P_s and P_t based on the Markov chain model
5: $T_{\text{avg}} = P_s T_s + P_t T_t$
6: MSR$(s, t) = P_s L_{\text{data}}(s)/T_{\text{avg}}$

4.5.2 Calculation of MSR with Multiple Contending Stations

Algorithm 3 estimates the MSR of station s given that a set of stations $\vec{\mathbf{t}}$ compete with s for the same interface. The basic idea is to pair s with the first station in $\vec{\mathbf{t}}$, say x, and treat s and x together as a virtual node with an updated effective transmission duration. Then the virtual node is paired with the second station in $\vec{\mathbf{t}}$. The process goes on until all stations in $\vec{\mathbf{t}}$ have been considered. As we will show next, this simple approximation performs reasonably well in various scenarios.

Algorithm 3 Calculation of MSR$_s$ given that a set of stations $\vec{\mathbf{t}}$ contend with s.

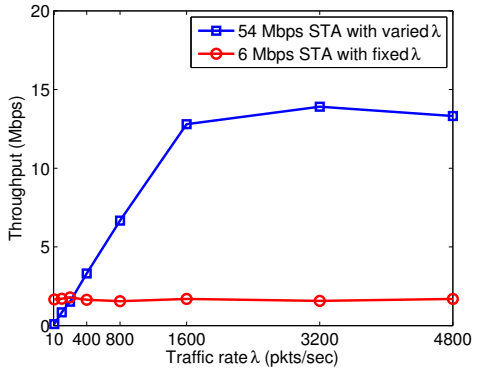
1: **INPUT:** $s, \vec{\mathbf{t}}$ and their status
2: **OUTPUT:** MSR$(s, \vec{\mathbf{t}})$
3: **for** each x in $\vec{\mathbf{t}}$ **do**
4: Run Algorithm 2 to calculate MSR$_{\text{tmp}}$ = MSR(s, x)
5: // Treat s and x as a virtual node with an updated T_s
6: Update $T_s = L_{\text{data}}(s)/\text{MSR}(s, x)$
7: Update $\vec{\mathbf{t}} = \vec{\mathbf{t}} \setminus \{s\}$
8: **end for**
9: MSR$(s, \vec{\mathbf{t}})$ = MSR$_{\text{tmp}}$

4.5.3 Validation of the Model and Estimation of G

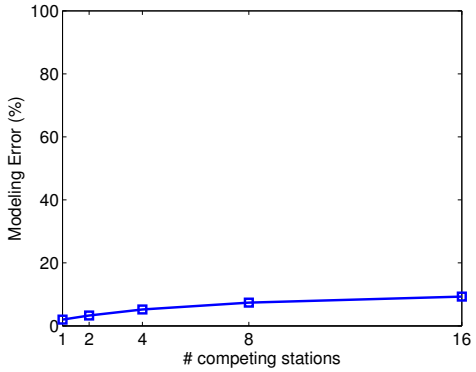
We use simulation results to validate the above model and algorithms for calculating the MSR. In the first simulation, there are two stations in the network. STA_1 has a PHY rate of 54 Mbps and STA_2 has a PHY rate of 6 Mbps. We fix the traffic rate of STA_2 to be 200 packets/s, where each packet carries 1500 bytes of data. We increase the traffic rate of STA_1 gradually. The throughput of two stations are plotted in Fig. 4.6(a). In this figure, we observe that the throughput of STA_1 first increases almost linearly with the traffic rate, and then keeps almost constant around 14 Mbps when the traffic rate is high. In other words, the MSR of STA_1 is 14 Mbps. Based on this observation, we approximate the throughput of a station as follows:

$$G = \min\{\text{MSR}, \lambda\}. \tag{4.6}$$

In comparison, our model gives an MSR of 13.83 Mbps, which is very close to the simulation result.



(a) Simulation results of two contending stations.



(b) Difference between simulated and analytical MSR results.

Figure 4.6 Simulation-based validation of the model and algorithms for calculating the MSR.

In the second simulation, we compare the simulated and analytical MSR results in a random setup. In this setup, we vary the number of competing stations, and each station chooses its PHY rate and traffic rate randomly. We plot the difference between the simulated and analyzed MSR results in Fig. 4.6(b). It shows that with a single competing station, our modeling scheme has an estimation error of about 2%. With 16 competing stations in the network, our model yields an estimation error of around 10% which is an acceptable tradeoff considering the simplicity of our model and low computational complexity of the algorithms.

4.6 Experimental Study

We have implemented SAP in MadWifi [20]. In this section, we evaluate its effectiveness using experimental results.

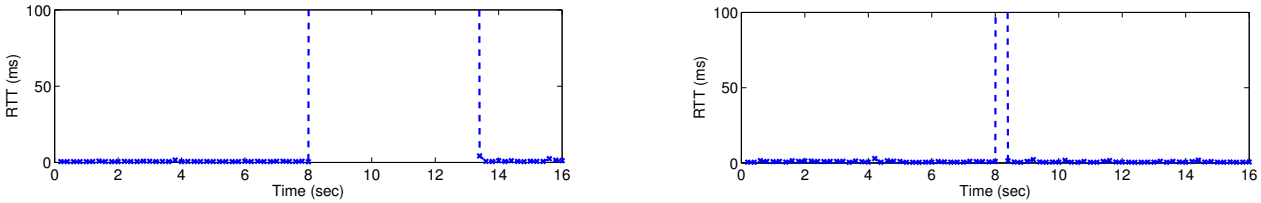
4.6.1 Experiment Setup

All the experiments are conducted with Dell desktops and laptops equipped various WLAN adapters, which all embed Atheros 5212 chipsets. The AP uses a Dell Optiplex GX520 desktop with two NetGear WAG311 PCI adapters. Clients use a Dell Latitude laptop equipped with one D-Link WNA1330 PCMCIA adapter. We use the off-the-shelf hardware instead of sophisticated equipments to conduct experiments as this makes our experimental results comparable to what users of commodity 802.11 devices may expect in realistic scenarios. We use Iperf [23] as the UDP packet generator to generate time-varying traffic loads. The association scheduling algorithm runs every 15 seconds in the SAP configuration.

We compare the performance of SAP against the following schemes: (i) the conventional OAOI scheme in which each AP only has a single interface; and (ii) a variant of the SAP scheme called SAP-ST which assumes all stations are saturated in the association scheduling algorithm, i.e., without considering the station's traffic load [37–39]. Note that, by placing two OAOI access points together and have them operate on different non-overlapping channels, it is equivalent to a basic OAMI access point with two interfaces. In this section, we first perform controlled experiments to evaluate the delay performance as well as the effectiveness of the load balancing algorithm. Then, we evaluate SAP in a random setup with more stations and more realistic traffic patterns.

4.6.2 Handoff Delay Performance

We first evaluate the delay performance of SAP with its seamless handoff architecture. In this experiment, we have a static SAP with two interfaces and one static station. The station keeps generating Ping packets (at an interval of 200 ms), which provides straightforward RTT measurements. At the 8-second mark, SAP switches the station to the other interface. Fig. 4.7(b) plots the RTT of the Ping packets. Recall that we have performed the same experiment under OAOI in Section 4.3.1.2; we re-plot



(a) Two OAOI access points.

(b) SAP access point with two interfaces.

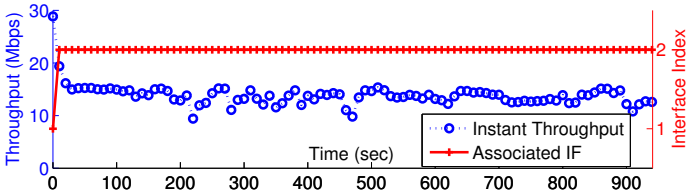
Figure 4.7 Comparison of handoff delay. Handoff starts at around the 8-second mark. Large round trip time indicates packet loss.

the results in Fig. 4.7(a). As shown in the figures, with SAP, the entire handoff delay is around 200 ms (for the station to perform the channel switch), which is significantly smaller than the six-second delay under OAOI. Note that, the station experiences one packet loss with SAP due to non-negligible channel switch time.

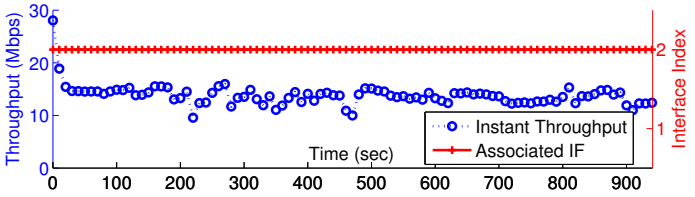
4.6.3 Load Balancing Performance

We then evaluate the load balancing performance of SAP with its smart association scheduling algorithm. In this experiment, we have one AP and three stations in the network. We manually set the PHY rate of each station as follows. Both STA_1 and STA_2 have a PHY rate of 48 Mbps while STA_3 has a PHY rate of 6 Mbps. All stations are receiving dowlink traffic with various traffic loads. STA_1 and STA_2 are heavily loaded ($\lambda \approx 30$ Mbps), while STA_3 has the following time-varying traffic pattern: $\lambda \approx 200$ Kbps between 0 and 300 seconds, 6 Mbps between 300 and 600 seconds, and 200 Kbps thereafter. STA_1 and STA_3 are initially associated with Interface 1, while STA_2 is associated with Interface 2 at the beginning. We collect the instant throughput (reported every 10 seconds), associated interface index and the TF value of each station, as well as the system aggregate throughput over run time. We compare SAP against SAP-ST in this experiment.

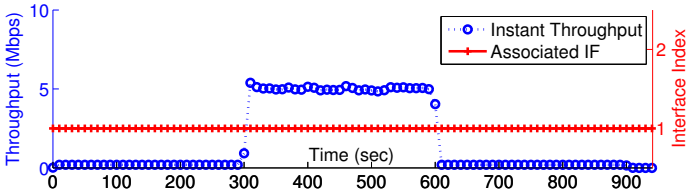
Performance of SAP-ST is shown in Fig. 4.8. SAP-ST uses saturation analysis without considering users' current traffic loads. Hence, SAP-ST schedules station-to-AP associations based only on the PHY rate of each station. As we can see in Figs. 4.8(a) (b) and (c), all three stations stay with the same interface throughout the evaluation period, even when STA_3 suddenly increases its traffic rate around the 300-second mark. With the saturation analysis, SAP-ST separates high-rate stations (i.e., STA_1 and STA_2 associated with Interface 2) from the low-rate station (i.e., STA_3 associated with Interface 1),



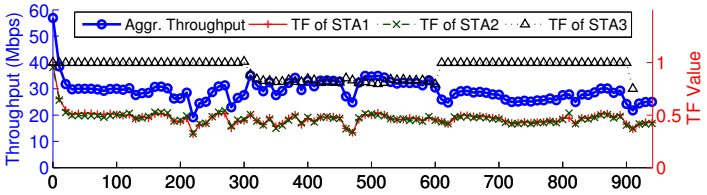
(a) Instant throughput and associated interface index for STA_1.



(b) Instant throughput and associated interface index for STA_2.



(c) Instant throughput and associated interface index for STA_3.

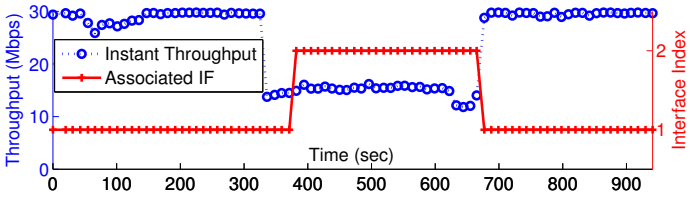


(d) Aggregate throughput and traffic fulfillment values.

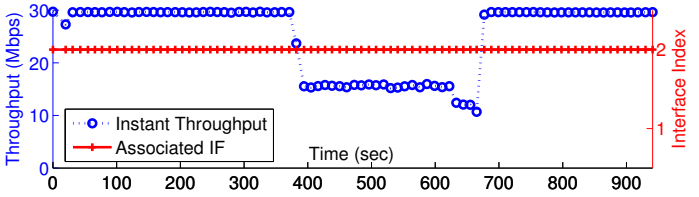
Figure 4.8 Throughput and fairness performances of SAP-ST. STA_1 and STA_2 have around 30 Mbps traffic rate. STA_3 has time varying traffic pattern: about 200 Kbps in [0 300) sec, 6 Mbps in [300 600) sec and 200 Kbps thereafter.

assuming that the performance anomaly caused by STA_3 always exists. With this association pattern, Interface 2 serves two heavily-loaded stations, but Interface 1 serves one station that is lightly-loaded for most of the run time, thus the loads on two interfaces are largely unbalanced. The aggregate system throughput and TF values are shown in Fig. 4.8(d). As we can see, SAP-ST actually satisfies STA_3 (i.e., $TF_3 \approx 1$) but sacrifices STA_1 and STA_2 (i.e., $TF_{1,2} \approx 0.5$). As a result, the total system throughput is only about 30 Mbps.

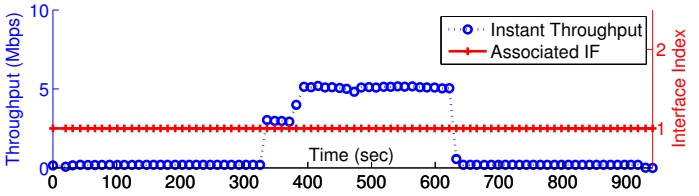
In comparison, the performance of SAP is shown in Fig. 4.9. By considering stations' traffic loads, SAP adjusts the association pattern dynamically. As we can see, one of the high-rate stations (i.e.,



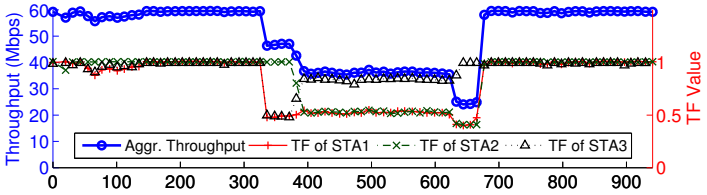
(a) Instant throughput and associated interface index for STA_1.



(b) Instant throughput and associated interface index for STA_2.



(c) Instant throughput and associated interface index for STA_3.



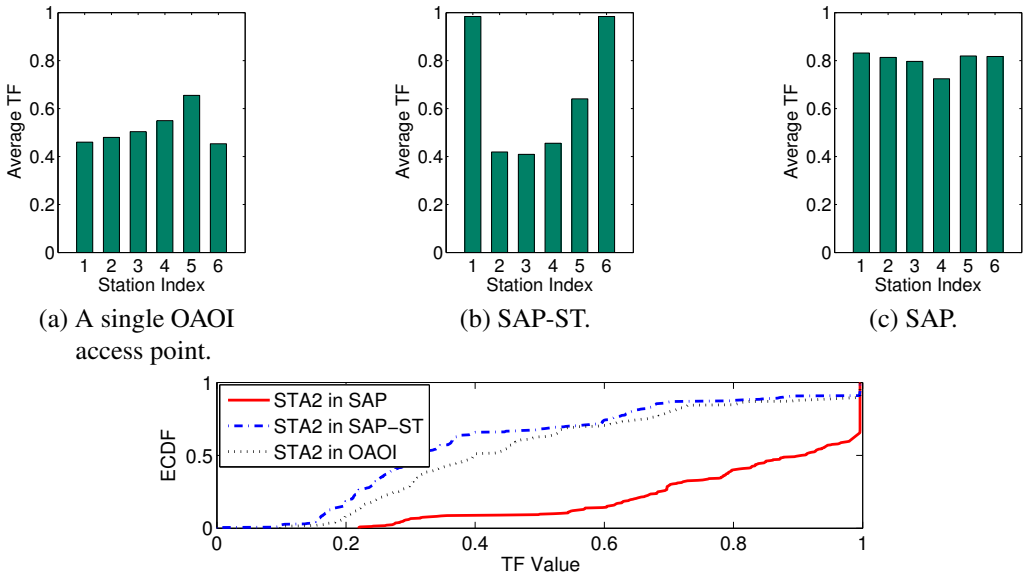
(d) Aggregate throughput and traffic fulfillment values.

Figure 4.9 Throughput and fairness performances of SAP under the same station setup as in Fig. 4.8.

STA_1) is associated with Interface 1 (same as STA_3) at the beginning when STA_3 is lightly-loaded hence no performance anomaly is observed. During this period, both STA_1 and STA_3 can be well satisfied and obtain a throughout close to the traffic rate requested. At the 300-second mark, STA_3 increases its traffic rate, due to which SAP smartly switches STA_1 to Interface 2 (same as STA_2) in order to achieve better satisfaction level for all stations. After STA_3 resumes its low traffic rate at the 600-second mark, SAP switches STA_1 back to Interface 1. In Fig. 4.9(d), we clearly see that the system throughout improves to around 60 Mbps for two thirds of the evaluation period, during which all stations are well served (i.e., $TF_{1,2,3} \approx 1$).

4.6.4 More Realistic Scenario

Lastly, we evaluate the performance of SAP in a more realistic scenario with six stations in the network, namely $\text{STA}_{1...6}$. In our setup, we positions six stations at different locations in our department building such that $r_4 \approx r_5 \approx 6$ Mbps, $r_2 \approx r_3 \approx 12$ Mbps, $r_6 \approx 24$ Mbps and $r_1 \approx 48$ Mbps. We uses Iperf to generate random downlink traffic to each station as follows. After every 75 seconds (on average) of idle time, Iperf randomly starts one of the following five traffic sessions: (i) Web browsing (30 seconds, about 30 Kbps bandwidth requirement); (ii) Audio (60 seconds, ∼100 Kbps); (iii) Video (10 minutes, ∼500 Kbps); (iv) HD Video (10 minutes, ∼6 Mbps) and (v) FTP (4 minutes, ∼20 Mbps). We run the experiments for one hour and record the TF value of each station during the entire experiment. We compare SAP against SAP-ST and a single OAOI access point in this scenario. Results are shown in Fig. 4.10.



(a) A single OAOI access point.

(b) SAP-ST.

(c) SAP.

(d) ECDF of TF values of STA_2.

Figure 4.10 Performance comparison in a more realistic scenario with six stations with various PHY rates and time-varying traffic rates.

Figs. 4.10(a), (b) and (c) plot the average TF value of each station during the experiment. From Figs. 4.10(a) and (b), we observe that without carefully coordinating the multiple interfaces, four out of six stations receive similar services in SAP-ST and OAOI, even though the number of interfaces has doubled in SAP-ST. Moreover, as shown in Fig. 4.10(b), SAP-ST favors high-rate stations (i.e., STA_1

and STA_6) which achieves a much larger TF value ($\sim$1.0) than low-rate stations ($\sim$0.4). The reason is that SAP-ST always discriminates low-rate stations due to performance anomaly, thus it groups high-rate stations and low-rate stations to different interfaces. As a result, high-rate stations can be well served by one interface while low-rate ones are crowded at another interface. In comparison, Fig. 4.10(c) shows that the TF values in SAP are more balanced than SAP-ST and much higher than OAOI. The minimum TF value in SAP is 0.72, in comparison to 0.41 in SAP-ST and 0.45 in OAOI. As an example to see how SAP provides better traffic fulfillment services to stations, we plot the ECDF of the TF values of STA_2 in Fig. 4.10(d). From the figure, we can see that STA_2 in SAP has TF $\geqslant 0.5$ for more than 90% of the time, while the portion is only 32% in SAP-ST and 37% in OAOI respectively.

4.7 Conclusions

From experiments, we observe that the handoff delay with legacy Wi-Fi APs is significant, and the severity of the performance anomaly in Wi-Fi networks varies with the traffic rates of connected clients. Based on these observations, we propose an advanced One-AP-Multiple-Interface (OAMI) architecture called SAP (Smart Access Point) to perform association scheduling from a unified AP and provide seamless handoff experience to clients. Additionally, we introduce a Traffic Fulfillment (TF) performance metric to aid in grouping stations during association scheduling. We have implemented SAP in the MadWifi device driver and demonstrated the effectiveness of SAP via experiments.

CHAPTER 5. MOBILITY-AWARE AND ADAPTIVE POWER MANAGEMENT

5.1 Introduction

Recent commercial success of Wi-Fi equipped portable devices such as smart phones opens a new era that users may rely on Wi-Fi networks to receive broadband services anytime and anywhere in various environments (e.g., static or mobile, indoor or outdoor, etc). Portable devices usually are battery-powered and have limited amount of energy. For example, the battery of most smart phones available in the market can only last a few hours under normal operations. So it is critical to prolong the battery lifetime for such portable devices. Wi-Fi radio has been known as a major source for sucking power out of portable devices. Like a Motorola Droid phone, battery life is cut in about half when the Wi-Fi radio is turned on. Hence, one of the effective way to prolong the battery lifetime is to implement an efficient power management scheme for Wi-Fi interfaces.

The 802.11 standard specifies a default power management scheme, called *Power Save Mode (PSM)*, which allows an 802.11 station to sleep for most of the time and only wake up periodically to receive the packets buffered at the Access Point (AP). The PSM was originally designed to reduce the power consumption of a wireless station when the station is lightly-loaded with delay-insensitive traffic. One of the limitations of the default PSM scheme is that it does not consider either user mobility or traffic condition when making the sleep/wakeup decisions. In other words, the station in PSM wakes up at every beacon interval and retrieves the buffered packets (if any) without considering its mobility pattern or the current traffic rate. In fact, more power-saving opportunities may emerge when these factors are considered. For example, by considering the user mobility pattern, more energy may be saved by buffering packets when the user is far away from the AP and transmitting them later at higher rates (hence less time and less energy consumption) when the user moves closer to the AP. Similarly, by considering the traffic condition, more energy may be saved by allowing the station to wake up less

frequently (hence less wakeup overheads) and retrieve all the buffered packets during a single beacon interval.

In this paper, we propose an enhanced PSM scheme, called M-PSM (Mobility-aware Power Save Mode), which exploits these additional power-saving opportunities. Specifically, M-PSM has the following features:

- M-PSM estimates the large-scale trend of the channel condition between a mobile station and the AP, based on which it buffers the packets whenever there is a chance for more energy saving.
- M-PSM monitors the traffic rate destined for a mobile station, based on which it adjusts the sleep interval for the station to reduce wakeup overheads.
- M-PSM avoids retrieving the buffered packets during deep fading dips by monitoring small-scale variations of the channel condition.
- We have implemented M-PSM in the MadWifi device driver and demonstrated its effectiveness via experiments.

The rest of the paper is organized as follows: in Section 5.2, we briefly explain the basic operations of the 802.11 PSM and discuss the related work; in Section 5.3, we present a few observations regarding the 802.11 PSM based on experiments and simulations; section 5.4 describes the design and implementation details of the proposed M-PSM scheme. Performance evaluation results are given in Section 5.5; and the paper concludes in Section 5.6.

5.2 Preliminary and Literature Survey

5.2.1 Power Save Mode (PSM) in 802.11 Networks

The 802.11 standard [1] specifies that an 802.11 station may be in one of the two power states: *active* or *doze*. In the active state, the station is fully powered and is ready to communicate with others at any time. In contrast, in the doze state, the station is powered down and consumes extremely low energy. However, it cannot transmit/receive packets or sense the wireless channel. Depending on how a station switches between active and doze states, there are two different operational modes for an

802.11 station: *Active Mode (AM)* or *Power Save Mode (PSM)*. AM Stations stay in the active state all the time, while PSM stations stay in the doze state for most of the time and only wake up at certain time instances to enter the active state to communicate with the AP. We are particularly interested in the PSM in this work.

An 802.11 station enters/leaves the PSM by transmitting data frames with the Power Management (PM) bit set to 1/0. The AP keeps track of the operational modes of all the stations that are associated with it. When a station is in PSM, the AP buffers the packets that are destined for the station and announces the buffer status every Target Beacon Transmission Time (TBTT) via the Traffic Indication Map (TIM) carried in the Beacon frame. If the TIM indicates the presence of buffered packets at the AP for a station, the station stays in the active state and sends PS-Poll messages to retrieve the buffered packets, one at a time, till all the packets are received; otherwise, it goes back to sleep in the doze state.

5.2.2 Related Work on PSM

Over the years, much work has been proposed to improve the performance of the 802.11 PSM. Unfortunately, little has been done to exploit how the user mobility may help improve PSM and how a PSM station shall behave when it is mobile.

In [45], the authors proposed to use a special hardware (a small radio device) to help reduce the energy consumption of an 802.11 station by allowing the station to stay in the deep sleep state and only waking it up when new data services for the station have been initiated by the AP. The authors of [46, 47] studied the tradeoff between minimizing energy consumption and reducing response delay with the 802.11 PSM. But, user mobility was not considered in their studies.

In [48–50], the authors proposed new protocols to minimize the energy consumption of a PSM station in the deep sleep state even when it moves across AP cell boundaries. Additionally, they found that, in order to retrieve the broadcast/multicast packets of its own interests, a PSM station has to receive the entire broadcast/multicast traffic from the AP, called *background traffic*, involuntarily since it cannot distinguish between packets *a priori* before the reception. Clearly, the presence of the background traffic may deteriorate the station's power saving efficiency. The authors of [51, 52] analyzed this issue in detail and suggested a new operation to mitigate the impact while maintaining the backward

compatibility with the 802.11 PSM. In [53], the authors proposed a scheduling policy to reorder traffic transmission priorities according to the operational states of destination stations. The key idea is that the AP transmits the traffic destined for PSM stations prior to the traffic for AM stations.

The proliferation of Wi-Fi equipped smart phones promotes the researches that target improving the power saving efficiency of an 802.11 PSM station when it is actively serving Voice over IP (VoIP) sessions [54–58]. In [59–61], the authors proposed to utilize proxies to aggregate traffic, thus reducing transmission overhead and energy consumption. Basically, the proxies reside in the AP, buffer the traffic destined for each PSM station until the buffered traffic could mitigate the transmission overhead, and then forward the buffered traffic with proper aggregation. The authors of [62] studied the relationship between user mobility and power consumption in cellular networks. It is a history-based approach and requires repetitive training before usage, hence is better for users traveling along known routes.

5.3 Observations & Motivations

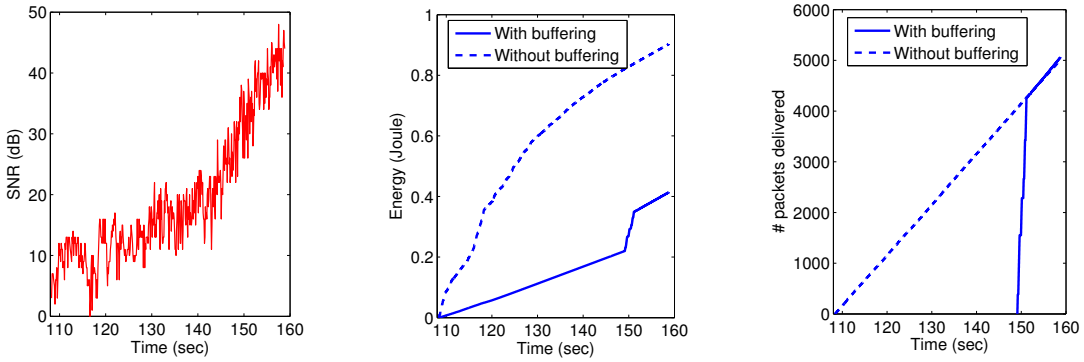
In this section, we present a few interesting observations regarding the 802.11 PSM from experiments and simulations, which motivate us to design our proposed scheme. Experiments are conducted with Dell Latitude laptops equipped with Linksys WPC55AG cards. Each laptop is loaded with the MadWifi device driver v0.9.4 [20] to collect experimental data. Simulations are conducted in Matlab using the traces collected from the experiments.

5.3.1 Effects of Buffering

Rate adaptation is a well-known technique to improve the throughput between an 802.11 station and the AP. The idea is to exploit the multiple transmission rates available for an 802.11 station and adjust its transmission rate dynamically to the time-varying and location-dependent channel condition between the station and the AP. With an effective rate adaptation scheme, a data packet may be transmitted at the highest rate that can be accommodated by the channel condition; hence less transmission time is required and less energy is consumed. This implies that the power saving efficiency of the default PSM scheme could be improved with a smart buffering strategy. The idea is that the AP buffers packets when the channel condition is bad and the station issues PS-Polls to retrieve the buffered packets only when

the channel condition has improved enough to accommodate higher transmission rates. Combining buffering with PSM could be particularly beneficial for mobile stations, which may observe large-scale fluctuations of the channel condition.

Fig. 5.1 plots the simulation results of PSM with a simple buffering strategy, with which the station issues PS-Polls only when the SNR is higher than 33 dB. Here, 33 dB is the minimum SNR value we found in experiments that data packets can be transmitted at the highest rate of 54 Mbps with negligible packet loss.



(a) SNR trace. (b) Cumulative energy consumption. (c) Total number of packets delivered.

Figure 5.1 Performance of PSM with a simple buffering strategy.

Fig. 5.1(a) shows the SNR trace used in the simulation, which is collected with a user carrying a laptop and walking towards the AP. Fig. 5.1(c) shows that, with the simple buffering strategy, the AP starts to deliver the buffered packets around the 149-second mark when the channel condition becomes very good. Comparing with the default PSM scheme, the extra buffering between the 109-second and 149-second marks yields 56% less energy consumption at the end of the simulation (i.e., 0.42 Joule with buffering vs 0.95 Joule without buffering at the 159-second mark). However, it should be noticed that such a simple buffering strategy might not work in other situations. For example, if the current SNR is lower than 33 dB but the channel condition is deteriorating (e.g., when the user is walking away from the AP), the buffered packets may never get a chance to be transmitted. So it is important to design a smart buffering strategy to work with PSM so that more energy may be conserved while all the packets are still delivered. The buffering strategy used in our proposed power-saving scheme will be discussed in Section 5.4.

5.3.2 Wakeup Overhead

To guarantee reception of the Beacon frames from the AP, an 802.11 station usually needs to wake up a bit earlier than the expected beacon arrival time. We call this lead time the *wakeup overhead*. The main reasons for the need of a wakeup overhead are that (i) the radio switching takes certain amount of time; and (ii) time asynchrony between the AP and the station due to clock drift. We have done some experiments to study the effects of the wakeup overhead. Fig. 5.2 plots the relationship between the wakeup overhead and the Beacon frame loss ratio for Linksys WPC55AG cards. Similar trend can be found for other Wi-Fi cards.

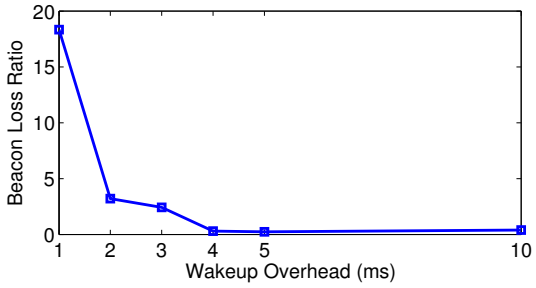


Figure 5.2 Effects of wakeup overhead on beacon reception for Linksys WPC55AG cards.

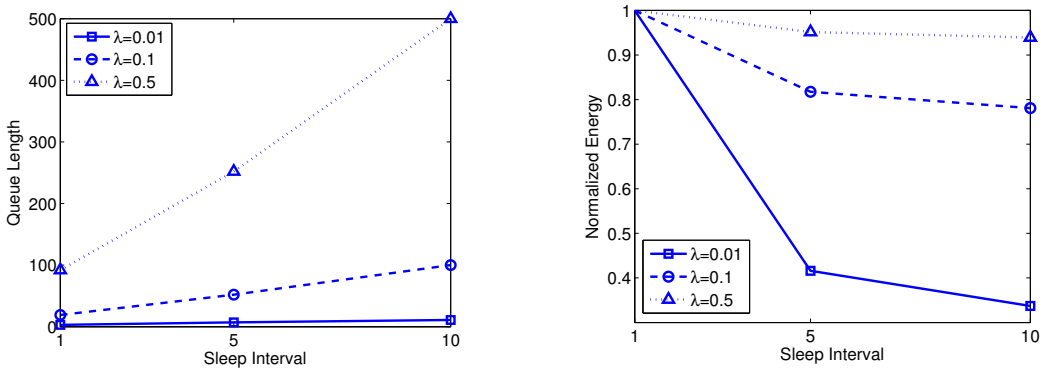
In general, the station experiences a higher beacon loss ratio with a smaller wakeup overhead. As shown in the figure, the beacon loss ratio could be as high as 18% when the wakeup overhead is only 1 ms. From the experiments, we can see that a minimum of 4 ms wakeup overhead is needed to guarantee a high beacon reception ratio. One way to reduce the overall wakeup overhead for an 802.11 station is that the station may skip some of the beacon announcements when the traffic is light, and wake up later to retrieve all the buffered packets from the AP with a single wakeup overhead.

5.3.3 Effects of Beacon Monitoring Interval (BMI)

As discussed above, an 802.11 station may skip some of the beacon announcements to conserve energy. We call the interval between two adjacent time instances when the station wakes up to receive beacon announcements as the *Beacon Monitoring Interval (BMI)*. For simplicity, we represent BMI in units of beacon intervals (i.e., 100 ms by default). So in a conventional 802.11 network, BMI = 1,

meaning that the station wakes up every beacon interval to listen for all beacon announcements from the AP.

We have performed trace-based simulations with different BMI values and results are plotted in Fig. 5.3. As shown in the figure, with a larger BMI, more packets are queued at the AP. As a result, the queue length increases and so does the packet delivery delay. On the other hand, more energy is conserved because more wakeup overheads are saved, particularly when the traffic rate is low (e.g., λ = 0.01 packets/ms). So there is a tradeoff in setting the proper BMI. We will discuss how BMI is adaptively adjusted in our proposed scheme in Section 5.4.



(a) Queue length at the AP.

(b) Normalized energy consumption.

Figure 5.3 Effects of Beacon Monitoring Interval (BMI). λ is the traffic rate (in units of packets/ms).

5.3.4 Design Guidelines

Based on the above observations, we conclude that an enhanced PSM scheme shall have the following features.

- It shall be able to predict the large-scale trend of the channel condition, and determine possible long-term buffering opportunities to fully utilize the high transmission rates in the future.
- It shall be able to detect possible short-term buffering opportunities to save the wakeup overhead, while guaranteeing that all the buffered packets can be delivered efficiently upon future wakeup.
- It shall be aware of the queue length as well as the traffic rate, and be able to adjust the BMI adaptively by considering these context information jointly.

5.4 M-PSM Design and Implementation

Following the design guidelines given in Section 5.3.4, we propose an enhancement to the default PSM scheme, called *M-PSM (Mobility-aware Power Save Mode)*. The highlights of the proposed M-PSM scheme are a smart buffering scheme and an adaptive BMI adjustment scheme. In the following, we first introduce the overall structure of M-PSM, and then describe the AP and station behaviors in detail respectively.

5.4.1 Overview

The overall structure of the proposed M-PSM scheme is shown in Fig. 5.4. In M-PSM, the AP buffers the incoming packets for all the stations that are currently associated with it and are in the power save mode. In the periodic Beacon frame sent by the AP, the AP announces the number of buffered packets for each station, in addition to the Traffic Indication Map (TIM) that only indicates the presence of the buffered packets. Upon reception of a PS-Poll frame from a station, the AP delivers a buffered packet to it. Details of the AP behaviors will be discussed in Section 5.4.2.

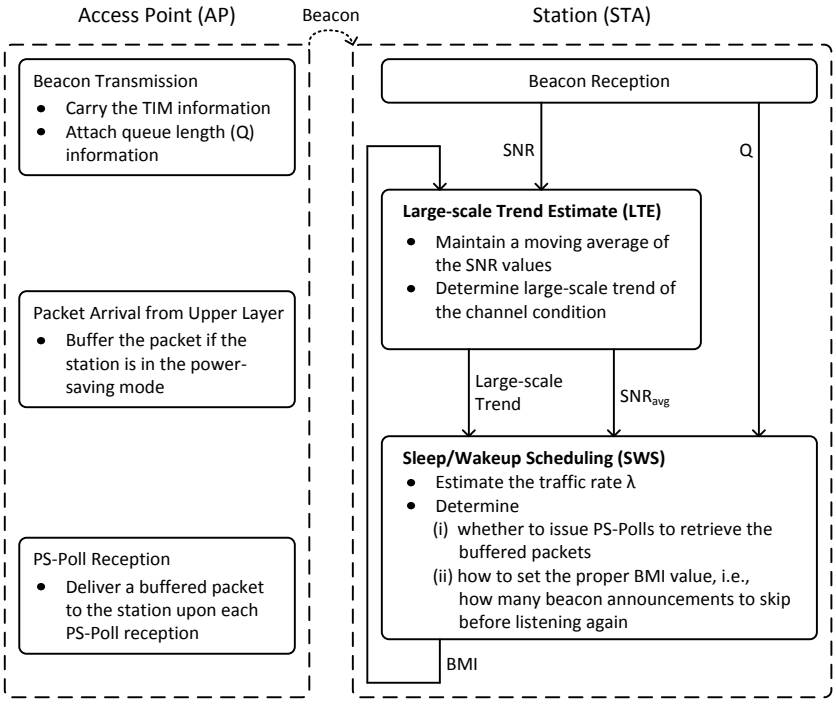


Figure 5.4 Overall structure of M-PSM.

In M-PSM, the station in the power save mode executes the following modules upon a beacon frame reception:

- *Large-scale Trend Estimation (LTE)*. This module takes the previously-calculated Beacon Monitoring Interval (BMI) and the SNR of the Beacon frame as inputs. The outputs are the moving average of the SNR values as well as the large-scale trend of the channel condition between the station and the AP, which could be "DOWN", "STABLE" or "UP". If the station is mobile, the large-scale trend could indicate the mobility pattern of the station. For example, a "DOWN" large-scale trend of the channel condition could imply that the station is moving away from the AP. Details of the LTE module will be discussed in Section 5.4.3.1.

- *Sleep/Wakeup Scheduling (SWS)*. This module accepts the outputs of the LTE module as well as the queue length information carried in the Beacon frame (i.e., the number packets buffered at the AP for the station and denoted by Q). Essentially, this module monitors the traffic rate λ destined for the station, and determines (i) whether to issue PS-Polls to retrieve the buffered packets; and (ii) how long to sleep before waking up for the next beacon announcement, i.e., the next BMI value. Details of the SWS module will be discussed in Section 5.4.3.2.

Before going any further, please note that M-PSM is an enhancement of the default PSM scheme, hence the application scenario for M-PSM is the same as PSM, which is when the station is lightly loaded with delay-insensitive applications. If the traffic is heavy or the application is delay-sensitive, neither PSM nor M-PSM should be used at all. As discussed later, M-PSM provides an option for upper layer applications to specify their delay requirements which are considered in the scheduling algorithm of M-PSM.

5.4.2 AP Behaviors

Similar to the default PSM, the AP in M-PSM buffers the packets for stations that are in the power save mode and deliver the buffered packets upon receiving PS-Poll messages from stations. The AP also indicates whether there is any packet buffered for the station via the TIM field in the periodic beacon announcement.

In addition, the AP in M-PSM attaches the queue length information in the Beacon frame. The queue length information is used by the station to estimate the traffic rate as well as to plan future sleep/wakeup schedules. In our implementation, we add one extra information element (using one of the reserved Element IDs) to the Beacon frame for this purpose. The format of the information element is shown in Fig. 5.5. For each station in the power save mode, its buffer queue length and its association ID (AID) are added to the information element. After receiving a Beacon frame, the station looks for its own AID in the information element to make sure that the correct queue length information is obtained. The 802.11 standard specifies that the maximum AID value is 2007. Hence we use 11 bits to contain the AID, and 13 bits to contain the queue length information (i.e., 8191 in maximum).

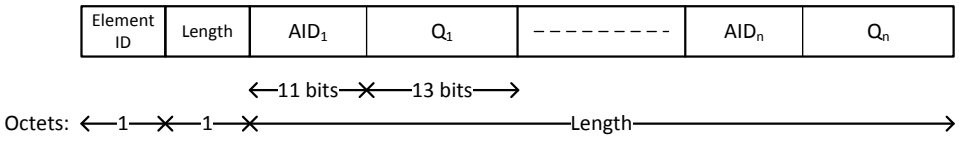


Figure 5.5 Format of the information element added to the Beacon frame in M-PSM. n is the number of stations that are in the power save mode.

5.4.3 Station Behaviors

In this section, we introduce the two M-PSM modules residing at the station side: the *Large-scale Trend Estimation (LTE)* module and the *Sleep/Wakeup Scheduling (SWS)* module.

5.4.3.1 Large-scale Trend Estimation (LTE)

The LTE module maintains a moving average of the SNR values of the received Beacon frames, and determines the large-scale trend (LT) of the current channel condition between the station and AP.

1.1) Calculation of the SNR Moving Average. Let S^t_{curr} denote the SNR value of the Beacon frame received at time index t. The SNR moving average is calculated as:

$$S^t_{\text{avg}} = \alpha\left(\text{LT}, S^t_{\text{curr}}\right) \cdot S^{t-1}_{\text{avg}} + \left[1 - \alpha\left(\text{LT}, S^t_{\text{curr}}\right)\right] \cdot S^t_{\text{curr}}, \tag{5.1}$$

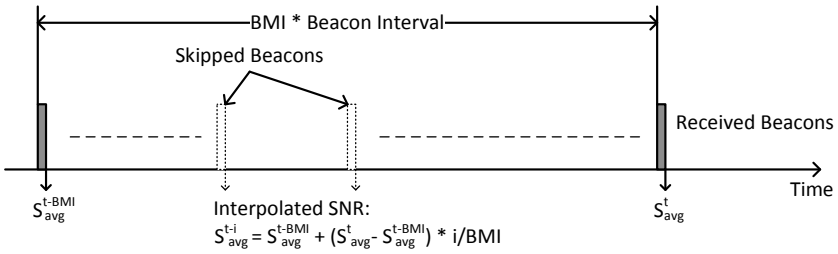


Figure 5.6 SNR interpolation in M-PSM.

where α is the smoothing factor:

$$\alpha\left(\mathrm{LT}, S^t_{\mathrm{curr}}\right) = \begin{cases} 0.8, & \text{if LT == DOWN or STABLE,} \\ \max\left\{0.8, 1 - 0.2 \cdot \frac{S^t_{\mathrm{curr}}}{33\ \mathrm{dB}}\right\}, & \text{if LT == UP.} \end{cases} \tag{5.2}$$

The reason for choosing such a smoothing factor is that, when the channel condition is stable or deteriorating, we would like to have S_{avg} track the channel variation more closely so that the station may retrieve the data packets whenever there is a chance; hence a relatively large weight (i.e., 0.2) is given to the new SNR sample. On the other hand, when the channel condition is improving, particularly when the current condition is poor, we would like to have a more conservative S_{avg} estimation which will later be used to predict the large-scale trend of the channel condition; an incorrect large-scale trend prediction could trigger improper buffering decisions and consequently unnecessary transmission delays and failures.

In M-PSM, the station may skip some of the beacon announcements, i.e., BMI > 1. In such cases, the moving averages of the SNR values at the skipped beacon times are estimated via interpolation. For example, as shown in Fig. 5.6, after skipping $(\mathrm{BMI} - 1)$ beacon announcements, the station receives a Beacon frame at time index t. Then, we calculate S^t_{avg} as

$$S^t_{\mathrm{avg}} = \alpha\left(\mathrm{LT}, S^t_{\mathrm{curr}}\right)^{\mathrm{BMI}} \cdot S^{t-\mathrm{BMI}}_{\mathrm{avg}} + \left[1 - \alpha\left(\mathrm{LT}, S^t_{\mathrm{curr}}\right)^{\mathrm{BMI}}\right] \cdot S^t_{\mathrm{curr}}. \tag{5.3}$$

and estimate S^{t-i}_{avg} $(1 \leqslant i \leqslant \mathrm{BMI} - 1)$ as:

$$S^{t-i}_{\mathrm{avg}} = S^{t-\mathrm{BMI}}_{\mathrm{avg}} + \left(S^t_{\mathrm{avg}} - S^{t-\mathrm{BMI}}_{\mathrm{avg}}\right) \cdot \frac{i}{\mathrm{BMI}}, \tag{5.4}$$

This equation will be used next in the pairwise comparison for estimating the LT.

1.2) Pairwise Comparison. The key function of the LTE module is to estimate the large-scale trend of the channel condition between the station and the AP. This is done via simple pairwise comparisons of the previously-calculated or interpolated S_{avg} values. Specifically, suppose a Beacon frame is received at time index t, then the following ϕ comparisons will be performed:

$$\left(S_{\text{avg}}^{t} - S_{\text{avg}}^{t-\phi}\right),\ \left(S_{\text{avg}}^{t-1} - S_{\text{avg}}^{t-\phi-1}\right),\ \cdots,\ \left(S_{\text{avg}}^{t-\phi+1} - S_{\text{avg}}^{t-2\phi+1}\right).$$

If more than ϕ_u out of the ϕ comparisons are larger than 0, we say that LT at time index t is "UP". On the other hand, if less than ϕ_d out of the ϕ comparisons are smaller than 0, we say that LT at time index t is "DOWN". Otherwise, LT = STABLE. In M-PSM, we heuristically set $\phi = 10$, $\phi_u = 0.7\phi$, and $\phi_d = 0.3\phi$ based on experimental trials. Table 5.1 gives an example of pairwise comparison at time index t = 20.

Table 5.1 An Example of Pairwise Comparison in the LTE Module

time index t	1	2	3	4	5	6	7	8	9	10
S_{avg}^{t} (dB)	8.4	7.1	6.1	6.3	6.6	6.9	6.9	6.9	6.1	6.1
Interpolated?	N	N	N	N	N	N	N	N	N	N
time index t	11	12	13	14	15	16	17	18	19	20
S_{avg}^{t} (dB)	5.5	6.0	7.2	8.7	10.0	11.3	11.3	11.3	11.3	11.3
Interpolated?	N	N	N	N	Y	N	Y	Y	Y	N
Comparison	$-$	$-$	$+$	$+$	$+$	$+$	$+$	$+$	$+$	$+$
LT	8 of 10 comparisons are positive while $\phi_u = 0.7\phi = 7$; hence LT at t = 20 is "UP".									

At last, we show in Fig. 5.7 an example trace of outputs by the LTE module collected from one of the experiments. In the figure, the y-coordinates of the markers represent the S_{avg} values at the corresponding time instances. Different markers represent different large-scale trends. For example, at time index t = 9 and 16, we have S_{avg}^{9} = 6.1 dB and S_{avg}^{16} = 11.3 dB, LT = DOWN and UP, and BMI = 1 and 4, respectively. Note that we are more interested in the large-scale trend instead of the small-scale variations, which explains the reason that the example in Fig. 5.7 does not match the small-scale trend very well.

5.4.3.2 Sleep/Wakeup Scheduling (SWS)

In M-PSM, the SWS module collects the following information from the Beacon frame as well as the LTE module: (i) Q – the queue length information carried in the Beacon frame; (ii) S_{avg} – the

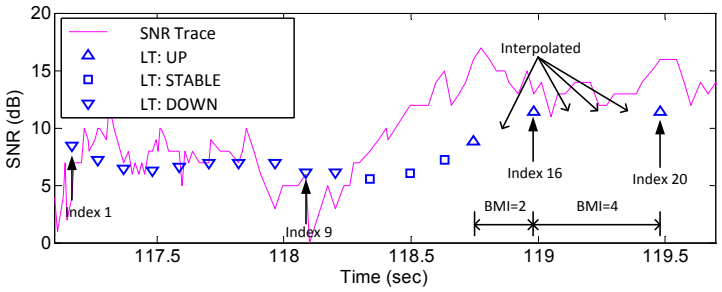


Figure 5.7 An example trace of outputs by the LTE module. The pairwise comparison for updating the LT at time index t = 20 is explained in Table 5.1.

moving average of the SNR values; and (iii) LT – the large-scale trend estimated by the LTE module. Based on the collected information, the SWS module performs the following functionalities:

- to estimate of the traffic rate destined for the station;
- to determine whether to issue PS-Polls immediately to retrieve the buffered packets;
- to decide the next BMI value, i.e., the number of beacon announcements to skip, if any.

The goal of the SWS module is to make sure that all the packets destined for the station are retrieved in more energy-efficient manners, i.e., at higher transmission rates with lower wakeup overheads.

The SWS module estimates the traffic rate (λ, in units of packets/ms) at time index t as follows:

$$\lambda^t = \beta \cdot \lambda^{t-1} + (1-\beta) \cdot \frac{\Delta Q}{\Delta T}, \tag{5.5}$$

where ΔQ and ΔT are increments of queue length and time since the last beacon reception, respectively. β is the smoothing factor and set to 0.9 to in M-PSM.

To determine whether to issue PS-Polls and how to set the next BMI value, the station exploits the following three types of opportunities. Upon each wakeup time, depending on the combination of the available opportunities, the station makes different decisions, as listed in Table 5.2. Next, we elaborate how each opportunity may be detected by the station and the actions taken by the station upon detection.

- *Sleep Opportunity:* Sleep opportunity is an opportunity that allows packet buffering in a relatively long term, in order to exploit the possibly better channel condition in the future. Sleep

Table 5.2 Decision Making in the SWS Module

Case	Conditions				PS-Poll?	How to set BMI?
1	$Q \geqslant 0.9 \cdot Q_{\text{limit}}$				Y	BMI = 1
2	$Q == 0$				N	BMI = $\min\{\text{BMI}_{\text{max}}, 2 \cdot \text{BMI}\}$
3	$0 < Q < 0.9 \cdot Q_{\text{limit}}$ && $S_{\text{avg}} \geqslant 33$ dB	nap opportunity			N	BMI = $\min\{\text{BMI}_{\text{max}}, \text{BMI}_{\text{nap}}\}$)
4		no nap opportunity			Y	BMI = 1
5	$0 < Q < 0.9 \cdot Q_{\text{limit}}$ && $S_{\text{avg}} < 33$ dB	LT == UP	sleep opportunity		N	BMI = $\min\{\text{BMI}_{\text{max}}, \text{BMI}_{\text{sleep}}\}$)
6			no sleep opportunity		Y	BMI = 1
7		LT == DOWN or STABLE	nap opportunity		N	BMI = $\min\{\text{BMI}_{\text{max}}, \text{BMI}_{\text{nap}}\}$)
8			no nap opportunity	small-scale opportunity	Y	BMI = 1
9				no small-scale opportunity	N	BMI remains unchanged

opportunity occurs when the following conditions are satisfied: (i) the AP can allow more packets to be buffered in the queue for the station. We define Q_{limit} as the maximum number of packets that can be buffered at the AP for a station, which is limited by the memory allocated by the AP to serve the station; (ii) the large-scale trend is "UP" so buffering packets could be beneficial since the station may be able to transmit the packets later at higher rates when the channel condition improves. (iii) the channel condition could be further improved in the near future. In our experiments, 33 dB is the minimum SNR value we found in experiments that accommodates transmissions of data packets at the highest rate of 54 Mbps with negligible packet loss. Hence, there is no need for buffering if $S_{\text{avg}} \geq$33 dB. To determine the sleep duration, the station needs to make sure that the available queue space at the AP allows more packets to be buffered, i.e., $\text{BMI}_{\text{sleep}} \geqslant 1$, where

$$\text{BMI}_{\text{sleep}} = \arg\max_{k} \left(Q + \lambda \cdot k \cdot \text{Beacon\_Interval} \leqslant 0.9 \cdot Q_{\text{limit}}\right). \tag{5.6}$$

Eq. (5.6) indicates that after sleeping for $\text{BMI}_{\text{sleep}}$ beacon intervals, the queue length at the AP will not exceed Q_{limit}. In practice, different applications might have different delay requirements. To handle this issue, M-PSM provides an option for the upper layer applications to specify their own Q_{limit} value. For example, the Skype might set a small value of Q_{limit} to avoid buffering for long time.

- *Nap Opportunity:* Nap opportunity is similar to sleep opportunity in that the station tries to exploit the chances to buffer more packets. However, the difference between them is that nap opportunities are shorter-term opportunities and the goal is to save the wakeup overhead. It may present itself only when the large-scale trend is "DOWN" or "STABLE", or when the channel

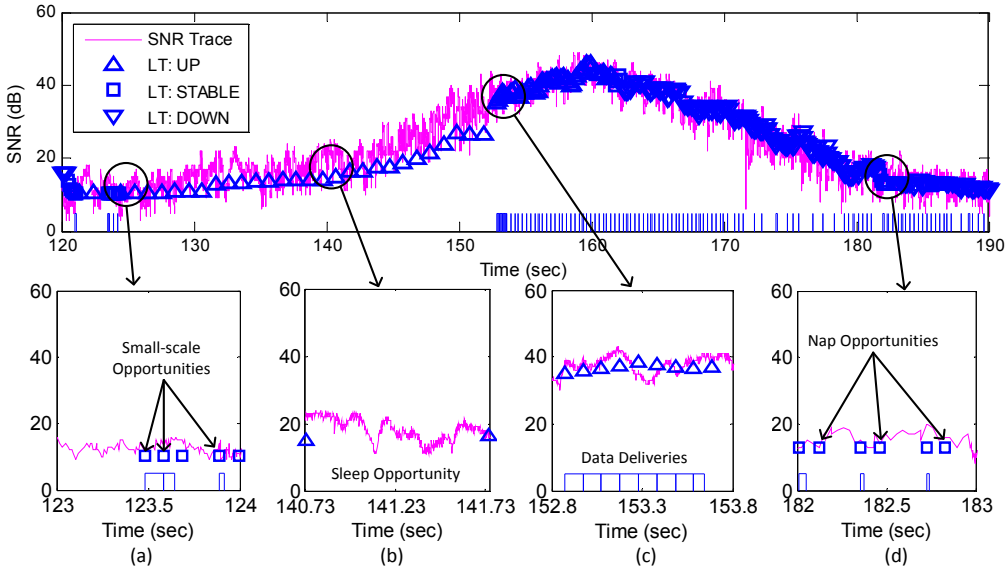


Figure 5.8 An example trace of outputs by the SWS module. (a) Example of small-scale opportunities. (b) Example of sleep opportunities. (c) Example of retrieving packets under excellent channel conditions. (d) Example of nap opportunities.

condition is already good enough (i.e., $S_{\text{avg}} \geqslant 33$ dB) to accommodate the highest transmission rate. In these situations, although it is unlikely that the station may be able to retrieve the buffered packets in the future at higher rates, more buffering could still allow the station to reduce the wakeup overheads by retrieving many packets together in one single wakeup. Comparing with sleep opportunity, nap opportunity usually is a much shorter-term phenomenon. When exploiting the nap opportunities, the station needs to make sure that all buffered packets could be transmitted during a single beacon interval, i.e., $\text{BMI}_{\text{nap}} \geqslant 1$, where

$$\text{BMI}_{\text{nap}} = \arg\max_{k} \left(Q + \lambda \cdot k \cdot \text{Beacon\_Interval} \leqslant 1 \text{ Mbps} \cdot \text{Beacon\_Interval} \right). \tag{5.7}$$

Eq. (5.7) guarantees that after sleeping for BMI_{nap} beacon intervals, all the buffered packets can be delivered within one single beacon interval, even at the lowest transmission rate of 1 Mbps. For this reason, nap opportunities usually occur when the traffic rate is low. Cases 3 and 7 in Table 5.2 show the actions taken by the SWS module when a nap opportunity is detected.

- *Small-scale Opportunity:* Different from the previous two, small-scale opportunity refers to the chances that the station should issue PS-Polls to retrieve the buffered packets. The goal is to, when there are no sleep or nap opportunities, avoid retrieving the packets during deep fading

dips, which usually are indicated by a smaller S_{curr} value than S_{avg}. In other words, the station only tries to retrieve packets when $S_{\text{curr}} \geqslant S_{\text{avg}}$, which is labeled as a small-scale opportunity. This can be seen from Cases 8 and 9 in Table 5.2.

There are two other cases in Table 5.2 that are not covered in the previous discussion: Case 1 when $Q \geqslant 0.9 \cdot Q_{\text{limit}}$ and Case 2 when $Q = 0$, which are straightforward to understand. Basically, if the queue is almost full, the station needs to retrieve immediately regardless of other conditions, while when there is no traffic for the station, the station could sleep longer by skipping more beacon announcements. In M-PSM, BMI_{max} is set to 10, meaning that the station needs to wake up at least once every 1 second to hear the beacon announcements from the AP and keep itself updated of the network/traffic conditions.

We show in Fig. 5.8 an example trace of outputs by the SWS module, as well as zoomed views at four different time periods. Y-coordinates of the markers represent the S_{avg} values at corresponding time instances. Short bars at bottom of the figures represent that the station is actively retrieving the packets. Sleep or nap opportunities can be observed clearly from (a), (b) and (d), where some of the beacon announcements are skipped to allow more packets to be buffered at the AP. In particular, as shown in (b), the station takes full advantage of the sleep opportunities by buffering all the packets which are transmitted later at higher rates when the channel condition becomes excellent around the 152.9-second mark. Comparing (b) and (d), we can see that nap opportunities are in a smaller scale than sleep opportunities. (a) shows how the SWS module exploits the small-scale opportunities. Upon receptions of the three Beacon frames pointed out in the figure, the station finds that no nap opportunities exist but the instant SNR is higher than the moving average. This means that small-scale opportunities appear and thus the station issues PS-Polls to retrieve packets after receiving these Beacon frames. (c) simply shows that when the channel condition is very good and the traffic rate is high, the station wakes up for every beacon announcement and retrieves as many packets as possible. Please note that Fig. 5.8 is just one example of the SWS module, and it does not necessarily imply that the data is buffered and delivered only when the SNR is larger than 33 dB. As discussed in Table 5.2, the transmission opportunities are not only related to the SNR but also the existences of sleep/nap/small-scale opportunities as well as the Q_{limit} value specified by the application.

5.5 Performance Evaluation

In this section, we evaluate the effectiveness of M-PSM using both experiments and trace-based simulations. Note that in M-PSM, the AP maintains a separate queue for each mobile station; the mobile stations make their own wakeup/sleep scheduling independently, and their actions do not affect the performances of each other. Hence, the performance of M-PSM is not affected by the number of stations in the network. For this reason, the performance evaluation in this work only involves an AP and a mobile station.

5.5.1 Experimental Study

5.5.1.1 Experimental Setup

We have implemented the proposed M-PSM scheme in MadWifi [20]. Hardware and software configurations used in our experiments are listed in Table 5.3. All the experiments are conducted with Dell Latitude E5400 laptops equipped with Linksys WPC55AG WLAN adaptors, which embed Atheros 5212 chipsets. We use off-the-shelf hardware instead of sophisticated equipments to conduct experiments as this makes our experimental results comparable to what users of commodity 802.11 devices may expect in realistic scenarios. The rate adaptation scheme used in our experiments is the RAM scheme proposed in [63], which is a fast-responsive rate adaptation scheme that represents the state of the art. We use a modified Iperf [23] as the UDP packet generator. CBR (Constant Bit Rate) traffic is generated with the packet size of 1000 octets. We set Q_{limit} to 800 in M-PSM.

Table 5.3 Configuration Parameters

Parameters	Values
Computer	Dell Latitude E5400 Laptop
Operating system	Linux Kernel 2.6.24-16
WLAN adaptor	Linksys WPC55AG
Device driver	MadWifi v0.9.4
PHY	IEEE 802.11g
Transmit Power	17 dBm
CBR packet size	1000 octets

We conduct experiments in both static and mobile scenarios. The experiment venue, the location of the static station as well as the trajectory of the mobile station are shown in Fig. 5.9. In order to min-

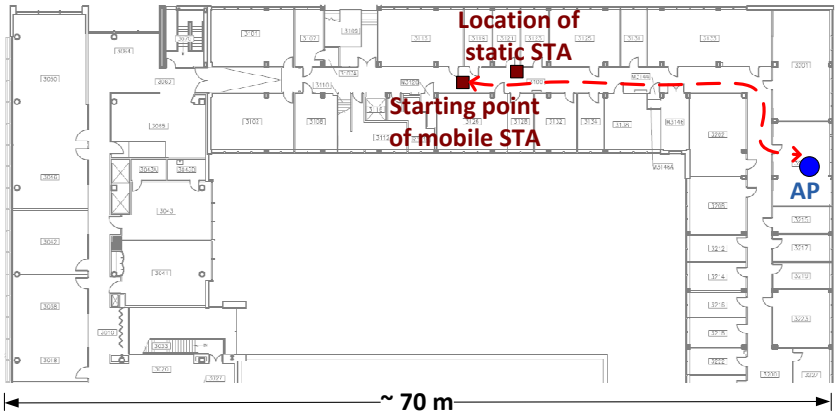
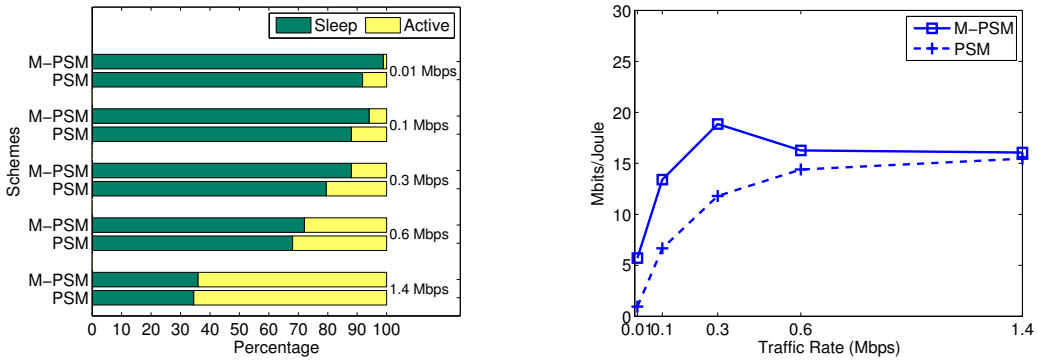


Figure 5.9 Experiment venue.

imize potential unexpected performance variation caused by people's movement and interference from other 802.11, ZigBee and Bluetooth devices, all experiments are conducted at nighttime or weekends. The result for each scenario is averaged over five experimental runs. We compare the performance of the proposed M-PSM scheme against the default 802.11 PSM scheme.

5.5.1.2 Mobile Scenario

In the mobile scenario, the station first moves toward the AP and then comes back to the starting point, as shown in Fig. 5.9, at a walking speed of about 1 m/s. In each experimental run, we measure the amount of time that the station sleeps in the doze state or stays awake in the active state. Fig. 5.10(a) compares the sleep/active time distributions of testing schemes under different traffic rates. Since the Linux kernel does not provide a direct measure of the energy consumption of Wi-Fi cards, we translate the measured time distribution to the energy-efficiency performance using typical Wi-Fi cards' power consumption rates that we find in a recent article [64]: 140 mW in the active state receiving data and 175 uW in the doze state sleeping. Fig. 5.10(b) compares the energy efficiency results. Since we are more interested in understanding how M-PSM saves energy via adjusting dynamically to the mobility pattern and traffic condition, rather than the exact amount of energy savings, this translation has limited impact on the observations to be presented next. From Fig. 5.10(b), percentage-wise non-negligible power saving could be observed when the traffic rate is less than 0.6 Mbps in the mobile case. We argue that this traffic rate is quite typical in practice, which shows the applicability of M-PSM.



(a) Sleep/active time distribution.

(b) Energy efficiency (Mbits/Joule).

Figure 5.10 Comparison of sleep/active time distributions and energy-efficiency performances under different traffic rates. Each point is averaged over five experimental runs.

Firstly, when the traffic rate is low, M-PSM allows the station to sleep for a much longer time than PSM. As shown in Fig. 5.10(a), when the traffic rate is 0.01 Mbps, the PSM station spends about 8.1% of the time in the active state, while the M-PSM station spends only about 1.1% of the time being active, thanks to M-PSM's capability of adjusting BMI dynamically to the traffic condition. When the traffic rate is low, M-PSM skips most beacon announcements to conserve energy. As a result, the energy efficiency of M-PSM is significantly higher than that of PSM (e.g., about 8 times higher when the traffic rate is 0.01 Mbps), as shown in Fig. 5.10(b).

Secondly, when the traffic rate is high (i.e., higher than 0.6 Mbps), the energy efficiency of M-PSM becomes stabilized to be around 16 Mbits/Joule. This is because the wireless link between the station and the AP becomes saturated when the traffic rate is high. Thus, there are few sleep or nap opportunities for the station, which spends most of the time retrieving the buffered packets.

Thirdly, as the traffic rate increases, the energy efficiency of M-PSM first increases, peaks when the traffic rate is about 0.3 Mbps, and then becomes stabilized. This interesting "peaking" phenomenon is unique and can only be observed in the mobile scenario for the following reason. During the station's movement towards the AP, sleep opportunities appear. Therefore, as the traffic rate increases, more packets may be buffered during each sleep opportunity and get transmitted later at much higher rates. On the other hand, as the traffic rate increases further, the packets get queued up quickly at the AP. As a result, the number of sleep opportunities decreases. The tradeoff between these two factors results in

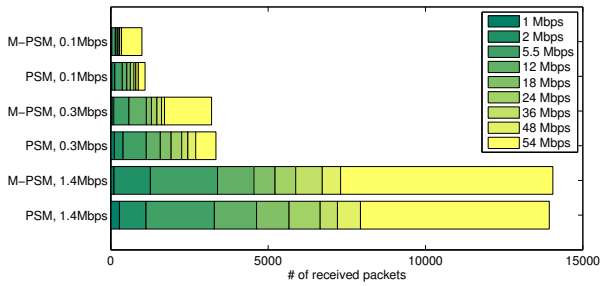
the energy efficiency of M-PSM peaking under medium traffic rates.

In order to have an in-depth understanding of the above tradeoff, we select an experimental run for each traffic rate in the mobile scenario, and compare their rate usage and sleep interval distributions in Fig. 5.11. Here, *sleep intervals* refer to the continuous time intervals during which the station stays in the doze state. Since BMI = 1 in the default PSM scheme, all the sleep intervals in PSM are around 100 ms or less, while with M-PSM, the station could exploit sleep and/or nap opportunities and longer sleep intervals are often observed. As shown in Fig. 5.11(b), when the traffic rate is low (i.e., 0.01 Mbps), most of the sleep intervals are longer than 100 ms with M-PSM. In fact, more than 80% of the sleep intervals are close to 1000 ms. This means that the station is able to identify the available sleep opportunities during the station's movement towards the AP. As a result, almost all the packets that arrive during the station's movement towards the AP are buffered to be transmitted later at the highest rate of 54 Mbps, as shown in the first horizontal bar in Fig. 5.11(a). In contrast, PSM blindly wakes up to listen for every single beacon announcement and retrieves the packets whenever the queue is nonempty. Therefore, its rate usage is more evenly distributed. On the other hand, when the traffic rate is high (i.e., 1.4 Mbps), very few sleep opportunities are available to the station and M-PSM performs similarly to the PSM, evidenced by the similar rate usage and sleep interval distributions between them.

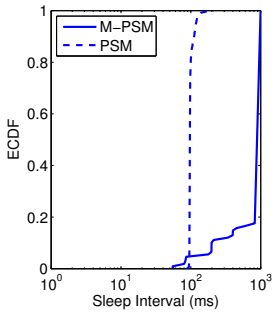
5.5.1.3 Static Scenario

In the static scenario, the station stays at a fixed location to communicate with the AP, as shown in Fig. 5.9. Sleep/active time distributions and energy-efficiency performances are compared in Fig. 5.12.

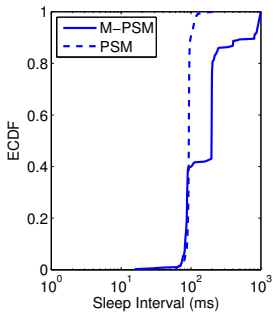
As expected, the "peaking" phenomenon, which was observed in the mobile scenario and caused by sleep opportunities, does not occur in the static scenario. This is because the channel condition between the static station and the AP is relatively stable; thus sleep opportunities are rarely seen by the station. On the other hand, similar to the mobile scenario, significant performance gain of M-PSM over PSM can be clearly seen under light traffic conditions (i.e., 0.01$\sim$0.1 Mbps), as shown in Fig. 5.12(a). This is due mainly to the nap opportunities identified. This is also confirmed by the significant difference between sleep interval distributions under light traffic conditions, as shown in Figs. 5.13(a) and (b).



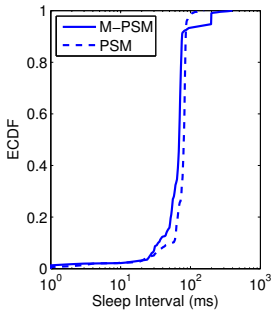
(a) Rate usage distribution.



(b) Sleep interval distribution (traffic rate is 0.01 Mbps).



(c) Sleep interval distribution (traffic rate is 0.3 Mbps).



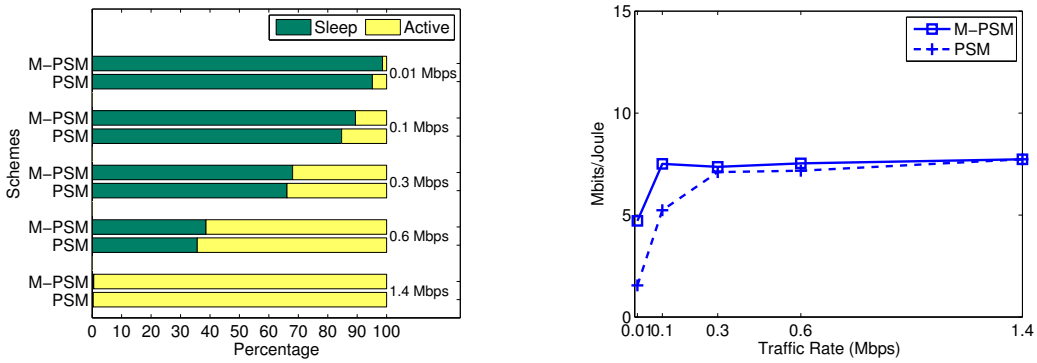
(d) Sleep interval distribution (traffic rate is 1.4 Mbps).

Figure 5.11 Comparison of rate usage and sleep interval distributions of three selected experimental runs in the mobile scenario.

5.5.2 Simulation Study

In this section, we perform trace-based simulations using Matlab to further evaluate the performance of M-PSM. Fig. 5.14(a) shows the SNR trace of one particular experimental run in the mobile scenario, based on which we simulate and compare the performances of M-PSM and PSM under the traffic rate of 0.1 Mbps. An SNR-to-transmission rate mapping table, which is obtained from experimental trials [38] and is consistent with [10], is used in the simulations to determine the proper transmission rates.

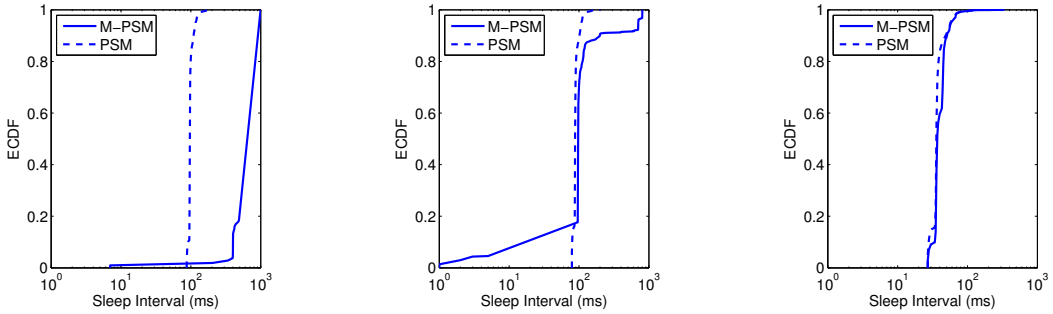
In Fig. 5.14(b), we plot the cumulative energy consumption over time. We can see that the energy consumption of M-PSM first increases in a slower pace than PSM since M-PSM is able to identify many sleep opportunities when the mobile station is moving towards the AP. Around the 150-second mark, there is a leap in the energy consumption for M-PSM. This is due to the fact that the channel condition becomes very good (higher than 33 dB, as shown in Fig. 5.14(a)) around that moment, when the station immediately starts issuing PS-Polls to retrieve the buffered packets. The behavior can also



(a) Sleep/active time distribution.

(b) Energy efficiency (Mbits/Joule).

Figure 5.12 Comparison of sleep/active time distributions and energy-efficiency performances under different traffic rates. Each point is averaged over five experimental runs.



(a) Sleep interval distribution (traffic rate is 0.01 Mbps).

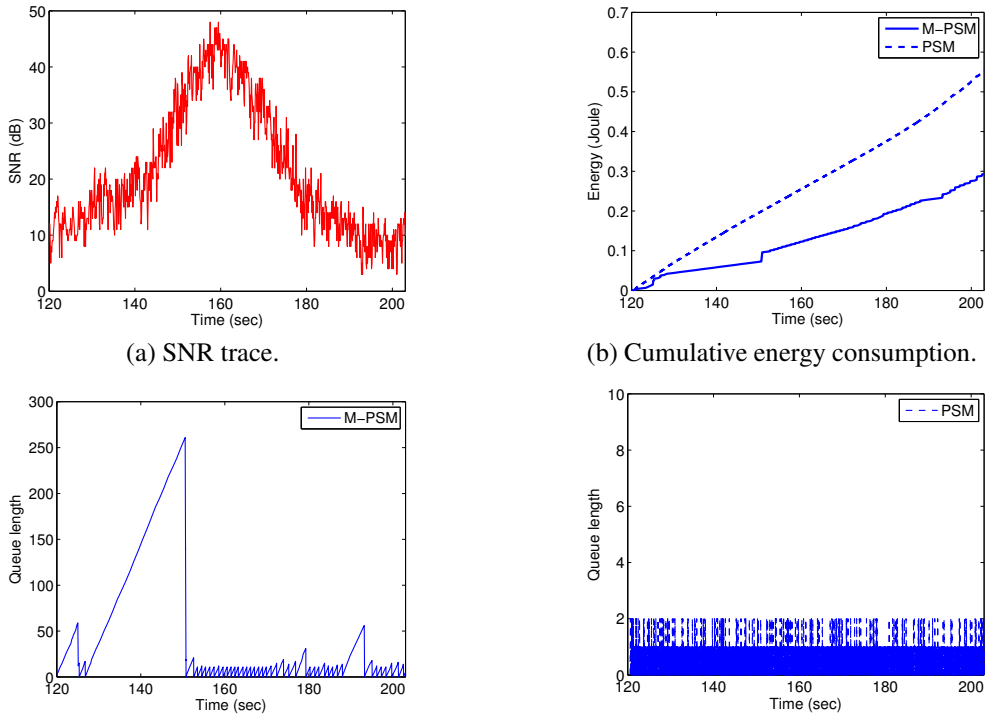
(b) Sleep interval distribution (traffic rate is 0.1 Mbps).

(c) Sleep interval distribution (traffic rate is 0.6 Mbps).

Figure 5.13 Comparison of sleep interval distributions of three selected experimental runs in the static scenario.

be observed from Fig. 5.14(c), where the queue length of M-PSM drops sharply around the 150-second mark because most of the buffered packets are delivered in a short period of time. After the 160-second mark, the mobile station moves away from the AP. Since then, very few sleep opportunities could be identified. Instead, M-PSM observes a few nap opportunities, which are determined mainly by the traffic condition rather than the mobility pattern. As a result, the station naps for a while (the queue length increases) and then wakes up to retrieve the buffered packets (the queue length decreases), yielding a sawtooth pattern of the queue length variation, as shown in Fig. 5.14(c).

Comparing with M-PSM, PSM requires a station to wake up every beacon interval to retrieve the buffered packets. Hence, the queue length of PSM is always small (never exceeding two as depicted in



(a) SNR trace.

(b) Cumulative energy consumption.

(c) Queue length of M-PSM.

(d) Queue length of PSM.

Figure 5.14 Comparison of cumulative energy consumption and queue length for M-PSM and PSM when the traffic rate is 0.1 Mbps.

Fig. 5.14(d)). Since PSM does not recognize or adjust to either mobility pattern or traffic condition, it consumes almost twice amount of energy as M-PSM.

5.5.3 Summary

We have the following observations from evaluation results:

- M-PSM inherits from PSM thus they have the same application scenarios which are only suitable for lightly-loaded stations with delay-insensitive applications running on top. In other scenarios, the PSM should not be used, neither should M-PSM. On top of that, M-PSM may save even more energy when the station's mobility pattern favors data buffering.

- When the traffic rate is low, M-PSM yields a much higher energy efficiency than PSM. This is due to M-PSM's strong ability to detect buffering opportunities and take proper actions upon detection.

- When the traffic rate is high, the performance improvement of M-PSM over PSM is limited, because there exist less buffering opportunities when more traffic is destined for the station. In fact, when the traffic rate becomes too high, the energy-efficiency performances of both M-PSM and PSM converge to that of the Active Mode.

- M-PSM reduces the power consumption of a station via exploiting various buffering opportunities. As a side effect, the packet transmission delay increases inevitably. Therefore, M-PSM is more applicable when the station runs delay-insensitive applications, e.g., file transfer or email. When the station runs delay-sensitive applications, e.g., Skype calls or multimedia streaming, special treatment may be needed if the station is to operate under the proposed M-PSM or the default PSM.

5.6 Conclusions

From experiments with the default 802.11 PSM, we find that (i) more energy may be saved for an 802.11 station by buffering packets at the AP under bad channel conditions and retrieve them later when the channel condition improves; and (ii) beacon reception by an 802.11 station incurs a non-negligible wakeup overhead. Motivated by these observations, we propose a mobility-aware power save mode, called M-PSM, which employs a smart buffering strategy and an adaptive sleep interval adjustment algorithm. We have implemented M-PSM in the MadWifi device driver. Experimental and simulation results show that M-PSM outperforms the default PSM scheme under both mobile and static scenarios, particularly when the traffic is light. Future work includes more rigorous studies of the effects of the power save mode on the delay performance of data delivery services and the extension of M-PSM with handoff considerations for mobile stations that roam between multiple APs.

CHAPTER 6. CONCLUSIONS AND POSSIBLE FUTURE TOPICS

In this chapter, we summarize the main contributions in this dissertation. Moreover, we discuss several potential research topics related to 802.11-based mobile networks based on our past research endeavors.

6.1 Research Contributions

In this dissertation, we have proposed several practical solutions to four critical resource adaptation issues in IEEE 802.11-based mobile networks. We have demonstrated their effectiveness via extensive experimental and simulation studies. The main contributions of our work are:

- *Prompt and Effective Rate Adaptation in Mobile Environments*

  In Chapter 2, we study rate adaptation in mobile environments. We propose a practical receiver-based rate adaptation scheme, called RAM (Rate Adaptation in Mobile environments) aiming at dealing with channel asymmetry and high fluctuation of channel conditions. Different from existing receiver-based rate adaptation schemes, RAM uses the variation of the ACK transmission rate to convey the feedback information implicitly. Also we adopt an effective scheme to guarantee that RAM-based and legacy 802.11 devices can interoperate with each other. We have implemented RAM with commercial 802.11 devices and the effectiveness of RAM is demonstrated in both mobile and static environments via experiments as well as simulations.

- *Fast and Seamless Handoff with Null Dwell Time*

  In Chapter 3, we study the handoff issue in mobile networks. We consider two fundamental challenges in the system design: (i) how to reduce the handoff delay; and (ii) how to make appropriate handoff decisions. We propose a practical fast handoff scheme, called HaND (Hand-

off with Null Dwell time). The key innovation of HaND is to leverage on the communication backbone between APs to relay the information about wireless channels as well as exchange other context information. Since the information is relayed over the backbone, the station does not need to dwell on the scanned channel to receive probe responses, hence the disruption to normal data services is minimized. Moreover, HaND adopts a novel satisfaction-based-fairness heuristic, which aims to provide fair service satisfaction to all stations and hence avoids possible performance anomaly caused by the transmission rate diversity. We have implemented it in the MadWifi device driver and demonstrated its effectiveness via experiments.

- *Seamless Load Balancing with Multiple Interfaces*

  In Chapter 4, we study the load balancing in 802.11 networks. We propose a unique solution called SAP (Smart Access Point) which utilizes a One-AP-Multiple- Interface (OAMI) architecture in order to (i) take full advantage of the popular OAMI architecture to provide seamless handoff experience to clients; (ii) smartly balance network load across multiple interfaces based on users' time-varying traffic load conditions. We also define a Traffic Fulfillment (TF) performance metric to quantify the user experience and aid in association scheduling. SAP is an AP-only solution and thus requires trivial network modifications while remaining backwards compatible with legacy 802.11 clients. We have implemented SAP in the MadWifi device driver and have demonstrated its effectiveness via experiments.

- *Mobility-aware Power Save Mode*

  In Chapter 5, we study the power save mode for 802.11 networks. In this work, we propose an enhancement to the default 802.11 Power Save Mode (PSM), called M-PSM, which exploits additional power-saving opportunities by considering user mobility and traffic condition when making the sleep/wakeup schedules for Wi-Fi interfaces. We have implemented M-PSM in the MadWifi device driver and demonstrated its effectiveness via experiments and trace-based simulations.

6.2 Possible Future Research Topics

The past research experiences greatly help us understand various practical problems in 802.11-based mobile networks. In this section, we share some of our opinions on these problems and discuss several potential research topics that are essential for future mobile networks.

- *Rate adaptation*
  - 802.11n is now the dominant Wi-Fi technology with its capability of MIMO communications. With MIMO technology, rate adaptation has one more adaptation dimension, i.e., the number of channel chains (streams), in addition to the modulation scheme. In the 802.11n case, rate adaptation has more flexibility. It needs more sophisticated rate adaptation scheme in order to fully exploit the advantage of MIMO technology in 802.11n. The future 802.11ac standard supports up to 8 streams which makes the problem even more challenging. Moreover, the 802.11ac standard specifies the usage of MU-MIMO which allows the AP to talk to multiple stations at the same time. This advanced feature actually raises more design challenges to rate adaptation scheme.
  - Different traffic type (access category) has a different QoS requirement. For example, video stream traffic might not care about the throughput but is more strict on the packet jitter, while best effort stream only care the total throughput. It would be possible for the rate adaptation scheme to take the different QoS requirements into account and deal with various traffic types differently.

- *Station handoff*
  - Most of the existing work on station handoff focuses on the MAC layer handoff without considering its effects on the upper layers. From the application's perspective, it would be important to maintain the TCP session upon a station handoff. Mobile IP could be one of the solutions in this case. However, since the handoff usually happens within the same subnet and APs are usually wired together, it might be possible to propose a simple but efficient solution in this scenario.

- *Load balancing*
  - As we have already discussed in the dissertation, the performance anomaly has to be considered in AP load balancing because the nature of the CSMA/CA MAC protocol in 802.11. The 802.11 DCF allows only one station to communicate with AP at one time, which is one of the causes of the performance anomaly (also a cause of the difficulty in supporting video/audio streams in WLANs). 802.11ac supports the AP to communicate with multiple stations simultaneously, which helps to alleviate the problem. Meanwhile, there are more considerations of load balancing with MU-MIMO, e.g., whether the station/AP supports MU-MIMO and how to group stations to better support MU-MIMO, etc..
  - IEEE802.11v (Wireless Network Management) standard specifies the protocol suites for load balancing in WLAN. However, the detailed algorithm is left to vendors, which might involve various research topics.

- *Power saving*
  - Various portable devices might adopt different power saving schemes to communicate with AP. Mobile stations might use PS-Poll, non PS-Poll or UAPSD as the power save scheme. In such a heterogenous environment, what might be the effects on the power saving performance? Will one power saving scheme affect the performance of another power saving scheme in such scenario?
  - Portable devices are mobile in nature, and they might be out of the range of all APs in certain cases, usually because of gaps between AP coverages. In this case, we would like the device to sleep as much as possible to save the energy while being able to wake up as soon as it enters the AP coverage area again. The tradeoff here is the connectivity versus energy. On one side, the device can keep awake to maintain the best connectivity. On the other side, the device will lose the chance to communicate with AP if it sleeps for too long.

- *Other topics*
  - Transmission power control (TPC), broadly speaking, is the intelligent selection of transmit power in the communication system to achieve good performance within the system. The goal of a good TPC scheme is to deliver packets successfully with less energy consumed which helps (i) prolong the battery life of the stations; and (ii) reduce the interference level in the network hence increasing the network capacity.
  - As Wi-Fi access points become increasingly prevalent in the urban environment, it is natural to ask if these access points may be exploited for purposes other than communication. In particular, can we make use of Wi-Fi signal strength information as a cue for localizing indoor mobile users? What accuracy can Wi-Fi achieve compared with other indoor localization schemes?

BIBLIOGRAPHY

[1] IEEE Standard 802.11-2007, *Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications, (Revision of IEEE Std 802.11-1999)*, 2007.

[2] A. Kamerman and L. Monteban, "WaveLAN-II: a high-performance wireless lan for the unlicensed band," in *Bell Labs Technical Journal*, vol. 2, no. 3, August 1997, pp. 118–133.

[3] M. Lacage, M. H. Manshaei, and T. Turletti, "IEEE 802.11 rate adaptation: a practical approach," in *ACM MSWiM'04*, 2004.

[4] J. Kim, S. Kim, S. Choi, and D. Qiao, "CARA: Collision-aware rate adaptation for IEEE 802.11 WLANs," in *IEEE Infocom'06*, 2006.

[5] S. Kim, S. Choi, D. Qiao, and J. Kim, "Enhanced rate adaptation schemes with collision awareness," in *IFIP Networking'07*, 2007.

[6] S. H. Y. Wong, H. Yang, S. Lu, and V. Bharghavan, "Robust rate adaptation for 802.11 wireless networks," in *ACM MobiCom'06*, 2006.

[7] J. Bicket, "Bit-rate selection in wireless networks," Master's thesis, MIT, 2005.

[8] G. Judd, X. Wang, and P. Steenkiste, "Efficient channel-aware rate adaptation in dynamic environments," in *ACM MobiSys'08*, 2008.

[9] J. del Prado Pavon and S. Choi, "Link adaptation strategy for IEEE 802.11 WLAN via received signal strength measurement," in *IEEE ICC'03*, 2003.

[10] J. Zhang, K. Tan, J. Zhao, H. Wu, and Y. Zhang, "A practical SNR-guided rate adaptation," in *IEEE Infocom Miniconference'08*, 2008.

[11] Onoe Rate Control, http://www.madwifi.org/browser/ trunk/ath_rate/onoe.

[12] G. Holland, N. Vaidya, and P. Bahl, "A rate-adaptive MAC protocol for multi-Hop wireless networks," in *ACM MobiCom'01*, 2001.

[13] B. Sadeghi, V. Kanodia, A. Sabharwal, and E. Knightly, "Opportunistic media access for multirate ad hoc networks," in *ACM MobiCom'02*, 2002.

[14] H. Jung, K. Cho, Y. Seok, T. Kwo, and Y. Choi, "RARA: Rate adaptation using rate-adaptive acknowledgment for IEEE 802.11 WLANs," in *IEEE CCNC'08*, 2008.

[15] C. Chen, H. Luo, E. Seo, N. H. Vaidya, and X. Wang, "Rate-adaptive framing for interfered wireless networks," in *IEEE Infocom'07*, 2007.

[16] Z. Li, A. K. Gupta, and S. Nandi, "A full auto rate (FAR) MAC protocol for wireless ad hoc networks," in *IEE Proceedings Communications*, vol. 153, no. 3, pp. 311–319, June 2005.

[17] P. Shankar, T. Nadeem, J. Rosca, and L. Iftode, "CARS: context aware rate selection for vehicular networks," in *IEEE ICNP'08*, 2008.

[18] D. Hadaller, S. Keshav, T. Brecht, and S. Agarwal, "Vehicular opportunistic communication under the microscope," in *ACM MobiSys'07*, 2007.

[19] Wistron CB9-GP-EXT 802.11a/b/g Cardbus Card, http://www.wneweb.com/products.htm.

[20] Multiband Atheros Driver for Wi-Fi, http://www.madwifi.org/.

[21] RSSI in Madwifi, http://madwifi-project.org/wiki/ UserDocs/RSSI.

[22] Atheros Registers, http://madwifi.org/wiki/DevDocs/ AtherosRegisters.

[23] Iperf, http://dast.nlanr.net/projects/Iperf.

[24] Network Simulator – ns-2, http://www.isi.edu/nsnam/ns/.

[25] A. Mishra, M. Shin, and W. Arbaugh, "An empirical analysis of the IEEE 802.11 MAC layer handoff process," in *ACM SIGCOMM Computer Communication Review*, vol. 33, no. 2, pp. 93–102, April 2003.

[26] V. Mhatre and K. Papagiannaki, "Using smart triggers for improved user performance in 802.11 wireless networks," in *ACM Mobisys'06*, 2006.

[27] M. Shin, A. Mishra, and W. Arbaugh, "Improving the latency of 802.11 handoffs using neighbor graphs," in *ACM Mobisys'04*, 2004.

[28] J. Teng, C. Xu, W. Jia, and D. Xuan, "D-Scan: Enabling fast and smooth handoffs in AP-dense 802.11 wireless networks," in *IEEE Infocom Miniconference'09*, 2009.

[29] H. Wu, K. Tan, Y. Zhang, and Q. Zhang, "Proactive Scan: Fast handoff with smart triggers for 802.11 wireless LAN," in *IEEE Infocom'07*, 2007.

[30] H. Velayos and G. Karlsson, "Techniques to reduce the IEEE 802.11b handoff time," in *IEEE ICC'04*, 2004.

[31] I. Ramani and S. Savage, "SyncScan: Practical fast handoff for 802.11 infrastructure networks," in *IEEE Infocom'05*, 2005.

[32] Y. Liao and L. Gao, "Practical schemes for smooth MAC layer handoff in 802.11 wireless networks," in *IEEE WoWMoM'06*, 2006.

[33] A. Mishra, M. Shin, and W. Arbaugh, "Context caching using neighbor graphs for fast handoffs in a wireless network," in *IEEE Infocom'04*, 2004.

[34] A. Giannoulis, M. Fiore, and E. Knightly, "Supporting vehicular mobility in urban multi-hop wireless netwroks," in *ACM MobiSys'08*, 2008.

[35] R. Murty, J. Padhye, R. Chandra, A. Wolman, and B. Zill, "Designing high performance enterprise Wi-Fi networks," in *USENIX NSDI'08*, 2008.

[36] M. Heusse, F. Rousseau, G. Berger-Sabbatel, and A. Duda, "Performance anomaly of 802.11b," in *IEEE Infocom'03*, 2003.

[37] Y. Bejerano, S. J. Han, and L. E. Li, "Fairness and load balancing in wireless LANs using association control," in *ACM MobiCom'04*, 2004.

[38] X. Chen and D. Qiao, "HaND: Fast handoff with null dwell time for IEEE 802.11 networks," in *IEEE Infocom'10*, 2010.

[39] W. Zhou and D. Qiao, "Fulfillment-based fairness: A new fairness notion for multi-AP wireless hotspots," in *IEEE ICC'07*, 2007.

[40] M. Laddomada, F. Mesiti, M. Mondin, and F. Daneshgaran, "On the throughput performance of multirate IEEE 802.11 networks with variable-loaded stations: Analysis, modeling, and a novel proportional fairness criterion," in *IEEE Transactions on Wireless Communications*, vol. 9, no. 5, pp. 1594–160, May 2010.

[41] T. Joshi, A. Mukherjee, Y. Yoo, and D. P. Agrawal, "Air time fairness for IEEE 802.11 multi rate networks," in *IEEE Transactions on Mobile Computing*, vol. 7, no. 4, pp. 513–527, April 2008.

[42] IEEE Standard 802.11h-2003, *Amendment 5: Spectrum and Transmit Power Management Extensions in the 5 GHz Band in Europe.*, 2003.

[43] D. Malone, K. Duffy, and D. J. Leith, "Modeling the 802.11 distributed coordication function in non-saturated heterogeneous conditions," in *IEEE/ACM Transactions on Networking*, vol. 15, no. 1, pp. 159–172, Feburary 2007.

[44] G. Bianchi, "Performance analysis of the IEEE 802.11 distributed coordination function," in *IEEE Journal on Selected Areas in Communications*, vol. 18, no. 3, pp. 535–547, March 2000.

[45] E. Shih, P. Bahl, and M. J. Sinclair, "Wake on Wireless: An event driven energy saving strategy for battery operated devices," in *ACM MobiCom'02*, 2002.

[46] R. Krashinsky and H. Balaknshnan, "Mimmizing energy for wireless web access with bounded slowdown," in *ACM MobiCom'02*, 2002.

[47] D. Qiao and K. G. Shin, "Smart power-saving mode for IEEE 802.11 wireless LANs," in *IEEE Infocom'05*, 2005.

[48] S. Jin, K. Han, and S. Choi, "A novel idle mode operation in IEEE 802.11 WLANs," in *IEEE ICC'06*, 2006.

[49] ——, "A novel idle mode operation in IEEE 802.11 WLANs: Prototype implementation and empirical evaluation," in *ACM WMASH'06*, 2006.

[50] ——, "Idle mode for deep power save in IEEE 802.11 WLANs," in *Journal of Communications and Networks*, vol. 12, no. 5, pp. 480–491, October 2010.

[51] Y. He and R. Yuan, "A novel scheduled power saving mechanism for 802.11 wireless LANs," in *IEEE Transactions on Mobile Computing*, vol. 8, no. 3, pp. 1368–1383, October 2009.

[52] Y. He, R. Yuan, X. Ma, and J. Li, "Analysis of the impact of background traffic on the performance of 802.11 power saving mechanism," in *IEEE Communications Letters*, vol. 13, no. 3, pp. 164–166, March 2009.

[53] E. Rozner, V. Navda, R. Ramjee, and S. Rayanchu, "NAPman: Network-assisted power management for WiFi devices," in *ACM MobiSys'10*, 2010.

[54] J. Liu and L. Zhong, "Micro power management of active 802.11 interfaces," in *ACM MobiSys'08*, 2008.

[55] Y. Chen, N. Smavatkul, and S. Emeott, "Power management for VoIP over IEEE 802.11 WLAN," in *IEEE WCNC'04*, 2004.

[56] B. Gleeson, D. Picovici, R. Skehill, and J. Nelson, "Exploring power saving in 802.11 VoIP wireless links," in *ACM IWCMC'06*, 2006.

[57] Y. Agarwal, R. Chandra, A. Wolman, V. Bahl, K. Chin, and R. Gupta, "Wireless wakeups revisited: Energy management for VoIP over Wi-Fi smartphones," in *ACM MobiSys'07*, 2007.

[58] V. Namboodiri and L. Gao, "Energy-efficient VoIP over wireless LANs," in *IEEE Transactions on Mobile Computing*, vol. 9, no. 4, pp. 566–581, April 2009.

[59] F. R. Dogar, P. Steenkiste, and K. Papagiannaki, "Catnap: Exploiting high bandwidth wireless interfaces to save energy for mobile devices," in *ACM MobiSys'10*, 2010.

[60] M. C. Rosu, C. M. Olsen, C. Narayanaswami, and L. Luo, "PAWP: A power aware web proxy for wireless LAN clients," in *IEEE WMCSA'04*, 2004.

[61] G. Anastasi, M. Conti, E. Gregori, and A. Passarella, "802.11 Power-saving mode for mobile computing in Wi-Fi hotspots: limitations, enhancements and open Issues," in *Wireless Networks*, vol. 14, no. 6, pp. 745–768, December 2008.

[62] A. Schulman, V. Navda, R. Ramjee, N. Spring, P. Deshpande, C. Grunewald, K. Jain, and V. N. Padmanabhan, "Bartendr: A practical approach to energy-aware cellular data scheduling," in *ACM MobiCom'10*, 2010.

[63] X. Chen, P. Gangwal, and D. Qiao, "RAM: Rate adaptation in mobile environments," in *IEEE Transactions on Mobile Computing*, vol. 8, no. 12, May 2011.

[64] Wi-Fi power management: Road warriors beware, http://blogs.techrepublic.com.com/networking/?p=533, May 2008.